

ORTAÖĞRETİM  
BİLİŞİM TEKNOLOJİLERİ VE  
YAZILIM DERSİ

ÖĞRETİM PROGRAMI

2023



T.C. MİLLÎ EĞİTİM BAKANLIĞI



**İÇİNDEKİLER**

<b>1. MİLLÎ EĞİTİM BAKANLIĞI ÖĞRETİM PROGRAMLARI</b> .....	4
1.1. ÖĞRETİM PROGRAMLARININ AMAÇLARI .....	4
1.2. ÖĞRETİM PROGRAMLARININ PERSPEKTİFİ .....	5
1.2.1. DEĞERLERİMİZ .....	5
1.2.2. YETKİNLİKLER .....	5
1.3. ÖĞRETİM PROGRAMLARINDA ÖLÇME VE DEĞERLENDİRME YAKLAŞIMI .....	7
1.4. BİREYSEL GELİŞİM VE ÖĞRETİM PROGRAMLARI .....	8
1.5. SONUÇ .....	8
<b>2. BİLİŞİM TEKNOLOJİLERİ VE YAZILIM DERSİ ÖĞRETİM PROGRAMI'NIN UYGULANMASI</b> .....	9
2.1. ÖĞRETİM PROGRAMININ TEMEL FELSEFESİ VE GENEL AMAÇLAR .....	9
2.2. ALAN BECERİLERİ .....	10
2.2.1. BİLGİ İŞLEMSEL DÜŞÜNME BECERİLERİ .....	11
2.2.2. BİLİMSEL SÜREÇ BECERİSİ .....	12
2.2.3. BİLGİ İŞLEMSEL SÜREÇ BECERİSİ .....	13
2.2.4. ÜST BİLİŞSEL SÜREÇ BECERİSİ .....	14
2.2.5. VERİ TANIMA, İŞLEME, ANALİZ BECERİSİ .....	15
2.2.6. 21. YÜZYIL BECERİLERİ .....	17
2.3. ÖĞRETİM PROGRAMININ UYGULANMASINDA DİKKAT EDİLECEK HUSUSLAR .....	19
2.4. ÜNİTE, KAZANIM SAYISI VE SÜRE TABLOSU .....	21
2.5. KİTAP FORMA SAYILARI VE KİTAP EBADI .....	22
<b>3. BİLİŞİM TEKNOLOJİLERİ VE YAZILIM DERSİ ÖĞRETİM PROGRAMI'NIN YAPISI</b> .....	23
3.1. ÖĞRETİM PROGRAMININ YAPISI .....	23
3.2. ÜNİTE, KAZANIM VE AÇIKLAMALARI .....	24

Bilim ve teknolojide yaşanan hızlı değişim, bireyin ve toplumun değişen ihtiyaçları, öğrenme öğretme teori ve yaklaşımlarındaki yenilik ve gelişmeler bireylerden beklenen rolleri de doğrudan etkilemiştir. Bu değişim; bilgiyi üreten, hayatta işlevsel olarak kullanabilen, problem çözebilen, eleştirel düşünen, girişimci, kararlı, iletişim becerilerine sahip, empati yapabilen, topluma ve kültüre katkı sağlayan vb. niteliklerdeki bir bireyi tanımlamaktadır. Bu nitelik dokusuna sahip bireylerin yetişmesine hizmet edecek öğretim programları salt bilgi aktaran bir yapıdan ziyade bireysel farklılıkları dikkate alan, değer ve beceri kazandırma hedefli, sade ve anlaşılır bir yapıda hazırlanmıştır. Belirtilen amaç doğrultusunda bir taraftan farklı konu ve sınıf düzeylerinde sarmal bir yaklaşımla tekrar eden kazanımlara ve açıklamalara, diğer taraftan bütünsel ve bir kerede kazandırılması hedeflenen öğrenme çıktıklarına yer verilmiştir. Her iki gruptaki kazanım ve açıklamalar ilgili disiplinlere göre hazırlanmıştır ve güncel, geçerli, eğitim öğretim sürecinde hayatla ilişki kurulabilecek nitelikler taşımaktadır. Bu kazanımlar ve bunların sınırlarını belirleyen açıklamalar, sınıflar ve eğitim kademeleri düzeyinde değerler, beceriler ve yetkinlikler perspektifinde bütünlük sağlayan bir bakış açısıyla yalın bir içeriğe işaret etmektedir. Böylelikle üst bilişsel becerilerin kullanımına yönlendiren, anlamlı ve kalıcı öğrenmeyi sağlayan, sağlam ve önceki öğrenmelerle ilişkilendirilmiş, diğer disiplinlerle ve günlük hayatla değerler, beceriler ve yetkinlikler çevresinde bütünleşmiş bir öğretim programları toplamı oluşturulmuştur.

### 1.1. ÖĞRETİM PROGRAMLARININ AMAÇLARI

Öğretim programları, 1739 sayılı Millî Eğitim Temel Kanunu'nun 2. maddesinde ifade edilen "Türk Millî Eğitiminin Genel Amaçları" ile "Türk Millî Eğitiminin Temel İlkeleri" esas alınarak hazırlanmıştır.

Eğitim ve öğretim programlarıyla sürdürülen tüm çalışmalar; okul öncesi, ilköğretim ve ortaöğretim seviyelerinde birbirini tamamlayıcı bir şekilde aşağıdaki amaçlara ulaşmaya yöneliktir:

1. Okul öncesi eğitimi tamamlayan öğrencilerin bireysel gelişim süreçleri göz önünde bulundurularak bedensel, zihinsel ve duygusal alanlarda sağlıklı şekilde gelişimlerini desteklemek
2. İlkokulu tamamlayan öğrencilerin gelişim düzeyine ve kendi bireyselliğine uygun olarak ahlaki bütünlük ve öz farkındalık çerçevesinde öz güven ve öz disipline sahip; gündelik hayatta ihtiyaç duyacağı temel düzeyde sözel, sayısal ve bilimsel akıl yürütme ile sosyal becerileri ve estetik duyarlılığı kazanmış; edimlerini etkin bir şekilde kullanan ve sağlıklı hayat yönelimli bireyler olmalarını sağlamak
3. Ortaokulu tamamlayan öğrencilerin, ilkokulda kazandıkları yetkinlikleri geliştirmek suretiyle millî ve manevi değerleri benimsemiş, haklarını kullanan ve sorumluluklarını yerine getiren, *Türkiye Yeterlilikler Çerçevesi'*nde ve ayrıca disiplinlere özgü alanlarda ifadesini bulan temel düzey beceri ve yetkinlikleri kazanmış bireyler olmalarını sağlamak
4. Liseyi tamamlayan öğrencilerin, ilkokulda ve ortaokulda kazandıkları yetkinlikleri geliştirmek suretiyle millî ve manevi değerleri benimseyip hayat tarzına dönüştürmüş, üretken ve aktif vatandaşlar olarak yurdumuzun iktisadi, sosyal ve kültürel kalkınmasına katkıda bulunan, *Türkiye Yeterlilikler Çerçevesi'*nde ve ayrıca disiplinlere özgü alanlarda ifadesini bulan temel düzey beceri ve yetkinlikleri kazanmış, ilgi ve yetenekleri doğrultusunda bir mesleğe, yükseköğretime ve hayata hazır bireyler olmalarını sağlamak.

## 1.2. ÖĞRETİM PROGRAMLARININ PERSPEKTİFİ

Eğitim sistemimizin temel amacı değerlerimiz ve yetkinliklerle bütünleşmiş bilgi, beceri ve davranışlara sahip bireyler yetiştirmektir. Bilgi, beceri ve davranışlar öğretim programlarıyla kazandırılmaya çalışılırken değerler ve yetkinlikler bu bilgi, beceri ve davranışların arasındaki bütünlüğü kuran bağlantı ve ufuk işlevi görmektedir. Değerlerimiz, toplumumuzun millî ve manevi kaynaklarından damıtılarak dünden bugüne ulaşmış ve yarınlarmıza aktaracağımız öz mirasımızdır. Yetkinlikler ise bu mirasın toplum hayatına ve insanlığın ortak kültürüne katılmasını ve katkıda bulunmasını sağlayan eylemsel bütünlüklerimizdir. Bu yönüyle değerlerimiz ve yetkinlikler birbirinden ayrılmaz bir şekilde teori-pratik bütünlüğündeki asli parçamızı oluşturur. Güncellik içinde öğrenme öğretme süreçleriyle kazandırmaya çalıştığımız bilgi, beceri ve davranışlar ise bizi biz yapan değerler ile yetkinliklerin günün şartları içinde görünürlük kazanma araç ve platformlarıdır; günün şartları içinde değişiklik gösterebilir yapısıyla arızidir ve bu sebeple de sürekli gözden geçirilerek güncellenir, yenilenir.

### 1.2.1. DEĞERLERİMİZ

Değerlerimiz öğretim programlarının perspektifini oluşturan ilkeler toplamıdır. Kökleri geleneklerimiz ve dünümüz içinde, gövdesi ve dalları bu köklerden beslenerek bugünümüze ve yarınlarmıza uzanmaktadır. Temel insani özelliklerimizi oluşturan değerlerimiz, hayatımızın rutin akışında karşılaştığımız sorunlarla başa çıkmada eyleme geçmemizi sağlayan kudretin ve gücün kaynağıdır.

Bir toplumun geleceğinin, değerlerini benimseyen ve bu değerleri sahip olduğu yetkinliklerle ete kemiğe büründüren insanlarına bağlı olduğu tartışma götürmez bir gerçektir. Bundan dolayı eğitim sistemimiz her bir üyesine uygun ahlaki kararlar alma ve bunları davranışlarında sergileme yeterliliğini kazandırma amacıyla hareket eder. Eğitim sistemi sadece akademik açıdan başarılı, belirlenmiş bazı bilgi, beceri ve davranışları kazandıran bir yapı değildir. Temel değerleri benimsemiş bireyler yetiştirmek eğitim sisteminin asli görevidir. Eğitim sistemi; yeni neslin değerlerini, alışkanlıklarını ve davranışlarını etkileyebilmektedir. Eğitim sistemi değerleri kazandırma amacı çerçevesindeki işlevini, öğretim programlarını da kapsayan eğitim programıyla yerine getirir. Eğitim programı; öğretim programları, öğrenme öğretme ortamları, eğitim araç gereçleri, ders dışı etkinlikler, mevzuat gibi eğitim sisteminin tüm unsurları göz önünde bulundurularak oluşturulur. Öğretim programlarında bu anlayışla değerlerimiz, ayrı bir program veya öğrenme alanı, ünite, konu vb. olarak görülmemiştir. Tam aksine bütün eğitim sürecinin nihai gayesi ve ruhu olan değerlerimiz, öğretim programlarının her birinde ve her bir biriminde yer almıştır.

Öğretim programlarında yer alan “kök değerler” şunlardır: *adalet, dostluk, dürüstlük, öz denetim, sabır, saygı, sevgi, sorumluluk, vatanseverlik, yardımseverlik*. Bu değerler, öğrenme öğretme sürecinde hem kendi başlarına hem ilişkili olduğu alt değerlerle hem de öteki kök değerlerle birlikte ele alınarak hayat bulacaktır.

### 1.2.2. YETKİNLİKLER

Eğitim sistemimiz yetkinliklerde bütünleşmiş bilgi, beceri ve davranışlara sahip karakterde bireyler yetiştirmeyi amaçlar. Öğrencilerin hem ulusal hem de uluslararası düzeyde kişisel, sosyal, akademik ve iş hayatlarında ihtiyaç duyacakları beceri yelpazeleri olan yetkinlikler *Türkiye Yeterlilikler Çerçevesi*'nde (TYÇ) belirlenmiştir. TYÇ'de sekiz anahtar yetkinlik belirlenmekte ve bunlar aşağıdaki gibi tanımlanmaktadır:

#### 1. OKUMA YAZMA YETKİNLİĞİ

Bu yetkinlik, diğer kişiler ile uygun ve yaratıcı bir yolla etkili iletişim ve bağlantı kurma yeteneğini ifade eder. Bireyler, çeşitli durumlarda sözlü ve yazılı iletişim kurma, izleme ve kendi iletişimlerini durumun gerekliliklerine

uyarlama becerilerine sahip olmalıdır.

Bu yetkinlik ayrıca farklı kaynak türlerini ayırt etme ve kullanma, bilgiyi araştırma, toplama ve işleme, aygıtları kullanma, sözlü ve yazılı görüşlerini bağlama uygun olarak ikna edici bir şekilde düzenleme ve ifade etme becerilerini de içerir.

## 2. ÇOKLU DİL YETKİNLİĞİ

Bu yetkinlik, farklı dilleri iletişim için uygun ve etkili olarak kullanma becerisini tarif eder. Dil yetkinlikleri, tarihî boyutu ve kültürler arası yetkinlikleri bütünleştirmektedir. Farklı diller ve iletişim araçları arasında geçiş yapma yeteneğine dayanır. Bireyin ihtiyaçlarına bağlı olarak farklı dillerde, farklı yeterlilik seviyelerinde sözlü mesajları anlama, sohbeti başlatma, sürdürme ve bitirme ile metinleri okuma, anlama ve yazma yeteneklerinden oluşur. Bireyler araçları uygun şekilde kullanabilmeli ve dilleri örgün, yaygın ve serbest olarak hayatı boyunca öğrenebilmelidir.

## 3. MATEMATİKSEL YETKİNLİK VE BİLİM, TEKNOLOJİ, MÜHENDİSLİKTE YETKİNLİK

**a. Matematiksel yetkinlik:** Matematiksel yetkinlik, günlük hayatta karşılaşılan problemi çözmek için matematiksel düşünme tarzını geliştirme ve uygulamadır. Sağlam bir aritmetik becerisi üzerine inşa edilen süreç, faaliyete ve bilgiye vurgu yapılmaktadır. Matematiksel yetkinlik, düşünme (mantıksal ve uzamsal düşünme) ve sunmanın (formüller, modeller, kurgular, grafikler ve tablolar) matematiksel modlarını farklı derecelerde kullanma becerisi ve isteğini içermektedir.

**b. Bilimde yetkinlik:** Bilimdeki yetkinlik, soruları tanımlamak ve kanıta dayalı sonuçlar üretmek amacıyla doğal dünyanın açıklanmasına yönelik bilgi varlığı ve metodolojiden yararlanma becerisi ve isteğine atıfta bulunmaktadır. Teknoloji ve mühendislikteki yetkinlik, insan istek ve ihtiyaçlarını karşılama bağlamında bilgi ve metodolojinin uygulanması olarak görülmektedir. Bilim, teknoloji ve mühendislikteki yetkinlik, insan etkinliklerinden kaynaklanan değişimleri ve her bireyin vatandaş olarak sorumluluklarını kavrama gücünü kapsamaktadır.

## 4. DİJİTAL YETKİNLİK

Bu yetkinlik öğrenme, çalışma ve topluma katılım için dijital teknolojileri güvenli, eleştirel, bilinçli kullanmayı ve bu teknolojilerle yakından ilgilenmeyi kapsamaktadır. Bilgi ve veri okuryazarlığı, iletişim ve iş birliği, medya becerileri, dijital içerik oluşturma (programlama dâhil), güvenlik (dijital refah ve siber güvenlikle ilgili yetkinlikler dâhil), fikrî mülkiyetle ilgili sorular, problem çözme ve eleştirel düşünmeyi içerir.

## 5. KİŞİSEL, SOSYAL VE ÖĞRENMEYİ ÖĞRENME YETKİNLİĞİ

Bu yetkinlik kendini yansıtma, zaman ve bilgiyi etkin bir şekilde yönetme, başkalarıyla yapıcı bir şekilde çalışma, esnek kalma ve kendi öğrenme ve kariyerini yönetme yeteneğidir. Belirsizlik ve karmaşıklıkla başa çıkabilmeyi, öğrenmeyi, fiziksel ve duygusal refahını desteklemeyi, fiziksel ve zihinsel sağlığını korumayı ve sağlık bilincine sahip, geleceğe yönelik bir yaşam sürmeyi, kapsayıcı ve destekleyici ortamda çatışmaya anlayış gösterme ve çatışmayı yönetme yeteneğini içerir.

## 6. VATANDAŞLIK YETKİNLİĞİ

Bu yetkinlik sosyal, ekonomik, yasal ve politik kavram ve yapıların yanı sıra küresel gelişmeler ve sürdürülebilirlik anlayışına dayalı olarak sorumlu vatandaş bilinciyle hareket etme, sivil ve sosyal hayata tam olarak katılma yeteneğidir.

## 7. GİRİŞİMCİLİK YETKİNLİĞİ

Bu yetkinlik, fırsatlar ve fikirler üzerinde hareket etme ve bunları başkaları için değerlere dönüştürme kapasitesi anlamına gelir. Yaratıcılık, eleştirel düşünme ve problem çözme, inisiyatif alma ve sebat etme, kültürel, sosyal veya finansal değeri olan projeleri planlamak ve yürütmek için iş birliği ile çalışabilme yeteneği üzerine kuruludur.

## 8. KÜLTÜREL FARKINDALIK VE İFADE YETKİNLİĞİ

Bu yetkinlik düşünce ve anlamların farklı kültürlerde, çeşitli sanatsal ve diğer kültürel formlarda nasıl yaratıcı bir şekilde ifade edildiğinin ve aktarıldığının anlaşılmasını ve saygı duyulmasını kapsamaktadır. Kendi düşünceleri ile toplumdaki rolüne ve yerine dair hislerini çeşitli ortamlarda ve farklı yollarla anlama, geliştirme ve ifade etme sürecine dâhil olmayı içerir.

### 1.3. ÖĞRETİM PROGRAMLARINDA ÖLÇME VE DEĞERLENDİRME YAKLAŞIMI

Hiçbir insan bir başkasının birebir aynısı değildir. Bu sebeple öğretim programlarının ve buna bağlı olarak ölçme ve değerlendirme sürecinin “herkese uygun”, “herkes için geçerli ve standart olması” insanın doğasına terstir. Bu sebeple ölçme ve değerlendirme sürecinde azami çeşitlilik ve esneklik anlayışıyla hareket edilmesi şarttır. Öğretim programları bu açıdan bir yol göstericidir. Öğretim programlarından ölçme değerlendirmeye ait bütün unsurları içermesini beklemek gerçekçi bir beklenti olarak değerlendirilemez. Eğitimde çeşitlilik; birey, eğitim düzeyi, ders içeriği, sosyal ortam, okul imkânları vb. iç ve dış dinamiklerden ciddi şekilde etkilendiği için ölçme ve değerlendirme uygulamalarının etkililiğini sağlamada öncelik öğretim programlarından değil öğretmen ve eğitim uygulayıcılarından beklenir. Bu noktada özgünlük ve yaratıcılık öğretmenlerden temel beklentidir.

Bu bakış açısından hareketle öğretim programlarında ölçme ve değerlendirme uygulamalarına yön veren ilkelere aşağıdaki gibi özetlemek mümkündür:

1. Ölçme ve değerlendirme çalışmalarında öğretim programının tüm bileşenleri ile azami uyum sağlanmalı, kazanım ve açıklamaların sınırları esas alınmalıdır.
2. Öğretim programı, ölçme sürecinde kullanılacak ölçme araç ve yöntemleri açısından uygulayıcılara kesin sınırlar çizmez, sadece yol gösterir. Ancak tercih edilen ölçme ve değerlendirme araç ve yönteminde gereken teknik ve akademik standartlara uyulmalıdır.
3. Eğitimde ölçme ve değerlendirme uygulamaları eğitimin ayrılmaz bir parçasıdır ve eğitim süreci boyunca yapılır. Ölçme sonuçları tek başına değil, izlenen süreçlerle birlikte bütünlük içinde ele alınır.
4. Bireysel farklılıklar gerçeğinden dolayı bütün öğrencileri kapsayan, bütün öğrenciler için genel geçer, tek tip bir ölçme ve değerlendirme yönteminden söz etmek uygun değildir. Öğrencinin akademik gelişimi tek bir yöntemle veya teknikle ölçülüp değerlendirilmez.
5. Eğitim sadece “bilme (düşünce)” için değil, “hissetme (duygu)” ve “yapma (eylem)” için de verilir; dolayısıyla ölçme ve değerlendirme çalışmalarında sadece bilişsel ölçümler yeterli kabul edilemez.
6. Çok odaklı ölçme değerlendirme esastır. Ölçme ve değerlendirme uygulamaları öğretmen ve öğrencilerin aktif katılımıyla gerçekleştirilir.
7. Bireylerin ölçme ve değerlendirmeye konu olan ilgi, tutum, değer ve başarı gibi özellikleri zamanla değişebilir. Bu sebeple söz konusu özellikleri tek bir zamanda ölçmek yerine süreç içindeki değişimleri dikkate alan ölçümler kullanmak esastır.

## 1.4. BİREYSEL GELİŞİM VE ÖĞRETİM PROGRAMLARI

Öğretim programlarının geliştirilmesi sürecinde insanın çok yönlü gelişimsel özelliklerine dair mevcut bilimsel bilgi ve birikim dikkate alınarak bütün bileşenler arasında ahengi dikkate alan harmonik bir yaklaşım benimsenmiştir. Bu bağlamda bazı temel gelişim ilkelerine değinmek yerinde olacaktır.

Öğretim programları, insan gelişiminin belirli bir dönemde sonlanmadığı ve gelişimin hayat boyu sürdüğü ilkesi ile hazırlanmıştır. Bu sebeple öğretim programlarında her yaş döneminde bireylerin gelişim özellikleri dikkate alarak destekleyici önlemler alınması önerilmektedir.

Gelişim, hayat boyu sürse de tek ve bir örnek yapıda değildir. Evreler hâlinde ilerler ve her evrede bireylerin gelişim özellikleri farklıdır. Evreler de başlangıç ve bitişleri açısından homojen değildir. Bu sebeple programlar bunu olabildiğince göz önünde bulunduran bir hassasiyetle yapılandırılmıştır. Programların amaçlarını ve kazanımlarını gerçekleştirme sürecinde gerekli uyarlamaların öğretmen tarafından yapılması beklenir.

Gelişim dönemleri ardışık ve değişmeyen bir sıra izler. Her evrede olup bitenler takip eden evreleri etkiler. Öte yandan bu ardışıklık basitten karmaşığa, genelden özele veya somuttan soyuta doğru gelişim gibi belirli yönelimlerle karakterize edilir. Program geliştirme sürecinde söz konusu yönelimler hem bir alandaki yeterliliği oluşturan kazanım ve becerilerin ön şart ve ardıllığı noktasında dikkate alınmış hem de sınıflar düzeyinde derslerin dağılımlarında ve birbirleriyle ilişkilerinde göz önünde bulundurulmuştur.

Öğretim programlarında insan gelişiminin bir bütün olduğu ilkesi ile hareket edilmiştir. İnsanın farklı gelişim alanlarındaki özellikleri birbirleri ile etkileşim hâlinindedir. Söz gelimi dil gelişimi düşünce gelişimini etkiler ve düşünce gelişiminden etkilenir. Bu sebeple öğretmenlerden öğrencinin edindiği bir kazanımın, gelişimde başka bir alanı da etkileyeceğini dikkate alması beklenir.

Öğretim programları bireysel farklılıklara ilişkin hassasiyetler göz önünde bulundurularak yapılandırılmıştır. Kalıtsal, çevresel ve kültürel faktörlerden kaynaklanan bireysel farklılıklar ilgi, ihtiyaç ve yönelme açısından da kendini belli eder. Öte yandan bu durum bireyler arası ve bireyin kendi içindeki farklılıkları da kapsar. Bireyler hem başkalarından farklılık gösterir hem de kendi içindeki özellikleri ile farklıdır. Örneğin bir bireyin soyut düşünme yeteneği güçlü iken aynı bireyin resim yeteneği zayıf olabilir.

Gelişim hayat boyu sürmekle birlikte bu gelişimin hızı evrelere göre değişkendir. Hızın yüksek olduğu zamanlar gelişim açısından riskli ve kritik zamanlardır. Bu sebeple öğretmenlerin, gelişim hızının yüksek olduğu zamanlarda öğrencinin durumuna daha duyarlı davranması beklenir.

## 1.5. SONUÇ

21. yüzyıldaki sosyal, ekonomik, siyasi ve teknolojik gelişmelere bağlı olarak ihtiyaç duyulan öğrenen niteliklerine ilişkin beklentiler de değişmiştir. Beklentilere cevap vermeye yönelik yeni dersler ihdas edilmiş ve öğretim programları beklentileri karşılayacak nitelikte zenginleştirilmiştir.

Öğrenenler yaşamlarında başarılı olabilmek için gerekli bilgi, beceri, değer ve tutumlara sahip olmalı ve hızla üretilen bilgi yığınları arasından gerekli ve doğru bilgiyi seçebilmelidir. Seçilen bilgileri analiz edip değerlendirerek elde etmeleri, elde ettikleri bilgiyi günlük yaşamlarında kullanabilmeleri için temel becerileri (genel ve alan becerileri) kazanarak yetkinlik sahibi olmaları önem arz etmektedir. Bu amaçla, istenen yetkinliklere sahip bireylerin yetişmesine hizmet edecek nitelikte hazırlanan öğretim programları salt bilgi aktaran bir yapıdan ziyade bireysel farklılıkları dikkate alan; bilgi, beceri, değer ve tutum kazandırma hedefli; sade ve anlaşılır bir yapıda hazırlanmıştır.



## 2.1. ÖĞRETİM PROGRAMININ TEMEL FELSEFESİ VE GENEL AMAÇLARI

Teknolojinin hayatımızdaki yeri, iletişimden öğrenmeye, iş dünyasından eğlenceye kadar geniş bir alanda kendini göstermektedir. Bilişim teknolojileri, bilgisayarlar ve elektronik cihazlar aracılığıyla verilerin işlenmesi, saklanması, iletilmesi ve alınmasını sağlayan araçlar ve yazılımları içermektedir. Bilişim teknolojilerinin temel amacı, verileri etkili bir şekilde kullanılabilir bilgiye dönüştürmektir.

Bu öğretim programı “İnsan, Toplum ve Bilim” alanında yer alan “bilişim teknolojileri ve yazılım” dersinde oluşturulan programlamaya giriş ve algoritma, robotik kodlama, programlama dilleri, mobil uygulama geliştirme, yapay zekâ uygulamaları modüllerinden oluşur.

Programlama Dilleri modülünde; Java, C++, C#, Dart, PHP, Javascript, Python programlama dillerinden birisi seçilir.

Programlamaya giriş ve algoritma modülü alınmadan; robotik kodlama, mobil uygulama geliştirme ve programlama dilleri modülleri seçilemez. Programlama dilleri modülü alınmadan yapay zekâ uygulamaları modülü seçilemez.

Yazılım ve algoritma kavramlarını öğrenmek, gençlerin problem çözme becerilerini geliştirmelerine yardımcı olabilir. Temel algoritma mantığı ve programlama bilgisi, öğrencilerin zorlu sorunlara analitik bir yaklaşım geliştirmelerine katkıda bulunabilir. Aynı zamanda gençlerin dijital dünyada kendilerine yer bulmalarını kolaylaştırabilir. Bu yetenekler, kişisel veya profesyonel projelerde kullanılacak değerli becerilerdir.

Bireyin bilişim teknolojileri yeterliliği kazanması, bu teknolojileri etkili bir biçimde kullanabilmesi zaruri bir ihtiyaçtır. Bu öğretim programı öğrencilere bilişim teknolojileri alanında genel yeterlilikleri ve becerileri kazandırmayı hedeflemektedir.

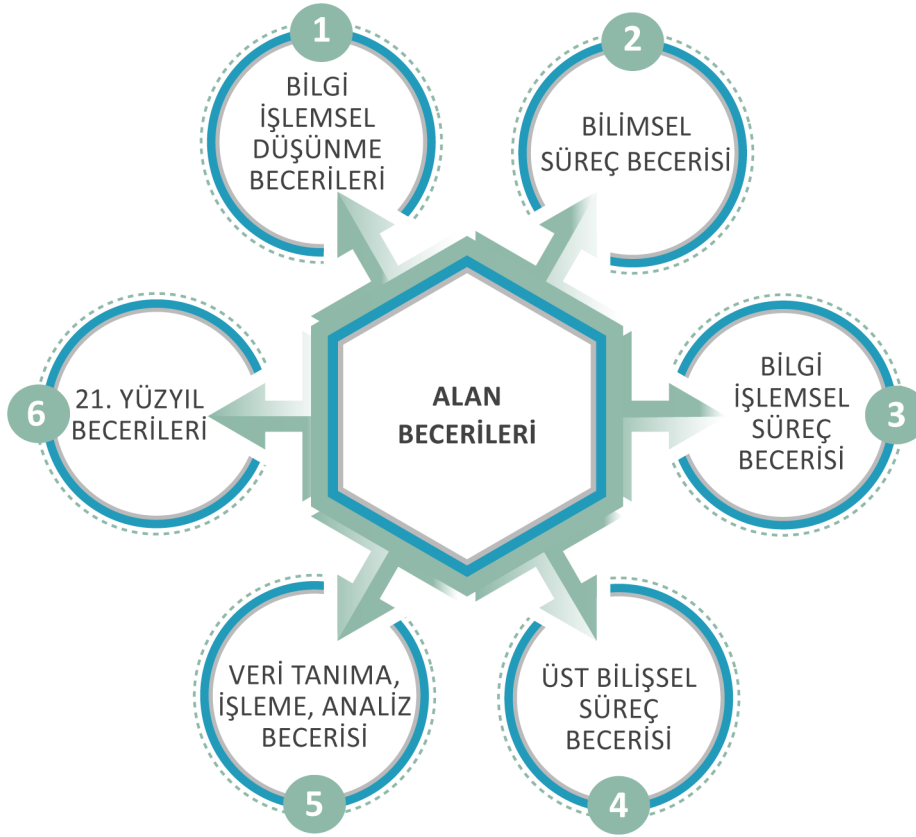
Bu programla öğrencilerin aşağıda belirlenen amaçlara ulaşmaları beklenmektedir:

- Teknoloji ile olan ilişkilerini gözden geçirerek tüketim ve üretim odağında rollerini belirlemeleri
- Teknolojiyi üretme, farkında olma ve kullanma süreçlerindeki sorumluluklarını anlamaları
- Proje veya ürün geliştirme sürecinde teknolojiyi etkin kullanmanın önemini farkında olmaları
- Yerli teknoloji üretiminin ülkemizin kalkınmasındaki rolünü ve önemini kavramaları
- Yerli ve milli teknoloji geliştirme konusundaki farkındalıklarının artması
- Teknoloji üretim süreçlerini anlamaları
- Yaşam boyu öğrenme becerilerini kazanmaları
- Bilişim teknolojileri becerilerini kullanarak proje oluşturabilmeleri ve ekip çalışması yürütebilmeleri
- Yerel kültürel değerlere saygı duymayı öğrenmeleri ve bu değerleri bilişim teknolojisi ve yazılım alanına yansıtılabilmeleri
- “Türkiye Yeterlilikler Çerçevesi” içerisinde ifadesi bulunan dijital yetkinliğe sahip bireyler olmaları
- Geleceğin teknolojilerine ve yönelimlerine hazırlıklı olarak kendilerini geliştirmeleri

- Üretken ve dijital vatandaşlar olarak yurdumuzun kalkınmasına katkı sağlayabilecek yazılımları geliştirebilmeleri
- Tasarım ve üretim yeteneklerini geliştirmeleri
- Teknoloji kullanarak özgün projeler veya ürünler geliştirmeleri

### 2.2. ALAN BECERİLERİ

Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı, bireylerin toplumsal ve mesleki hayatlarında başarılı olabilmeleri için gereken temel becerilere odaklanmaktadır. Bu beceriler, bilgiye erişim ve günlük yaşam problemlerinin çözümünde kullanma, karmaşık görevleri yerine getirme, iş birliği yapma, teknolojiyi etkili bir şekilde kullanma ve toplumsal olaylardan haberdar olma gibi çok yönlü yetenekleri kapsamaktadır.



Şema 1 : Bilişim Teknolojileri ve Yazılım Dersi Alan Becerileri

Günümüz koşullarında duyulan ihtiyaçlar göz önünde bulundurularak tasarlanan bu öğretim programı, öğrencilerin çağın gereksinimlerine uygun beceriler kazanmalarını hedeflemektedir. Bunların yanında bilişim teknolojileri ve yazılım dersinde kazandırılması gereken özel alan becerileri bulunmaktadır. Bu beceriler:

- Bilgi İşlemsel Düşünme Becerisi
- Bilimsel Süreç Becerisi
- Bilgi İşlemsel Süreç Becerisi

- Üst Bilişsel Süreç Becerisi
- Veri Tanıma, İşleme, Analiz Becerisi
- 21. Yüzyıl Becerileri olarak belirlenmiştir.

### **2.2.1. BİLGİ İŞLEMSEL DÜŞÜNME BECERİLERİ**

Bilgi işlemsel düşünme, bilişim teknolojileri ve yazılım dersinin temelini oluşturan bir düşünme sürecidir. Bu beceri, öğrencilere karmaşık problemleri tanımlama, analiz etme, çözme ve bilgisayar tabanlı araçlar kullanarak uygulama yeteneği kazandırır. Bilgi işlemsel düşünme, öğrencilerin bilişsel, özgün ve eleştirel düşünme becerilerini geliştirmek ve teknolojik problemlere etkili çözümler bulmalarını sağlamak için gereklidir. Bu süreç, öğrencilere problem tanımlama, veri analizi, algoritma geliştirme, problemleri çözme ve özgün yaklaşımlar geliştirme becerilerini kazandırır. Ayrıca, bilgi işlemsel düşünme, öğrencilere programlama yeteneği kazandırarak onları teknolojik dünyada etkili katılımcılar haline getirir. Bu beceri, modern dünyada bilgi işleme ve teknoloji kullanımı konularında bilinçli ve yetenekli bireylerin yetişmesine katkı sağlar.

- Soyutlama Becerisi
- Mantıksal Düşünme Becerisi
- Test Etme ve Hata Ayıklama Becerisi
- Algoritmik Düşünme Becerisi
- Genelleme Becerisi

### **2.2.2. BİLİMSEL SÜREÇ BECERİSİ**

Bilimsel süreç becerileri, öğrenmeyi kolaylaştıran, öğrencilerin aktif olmasını ve kendi öğrenmelerinin sorumluluklarını almasını sağlayan, öğrenmenin kalıcılığını artıran ve ayrıca araştırma yol ve yöntemlerini kazandıran temel becerilerdir. Bireyler bu becerileri günlük hayatın her aşamasında kullanabilirler. Bu becerileri kazanan öğrenciler gözlem yapmayı, tahmin etmeyi, ölçümlemeyi ve analiz etmeyi bilirler. Bu işlemler sonucu, sonraki becerilere zemin bilgi üretebilir, çözüm yolları önerebilir, çözüm önerilerini sınyarak bir sonuca ve nihayetinde çözüme ulaşabilirler.

- Gözlem Yapma Becerisi
- Tahmin Etme Becerisi
- Ölçüm Yapma Becerisi

### **2.2.3. BİLGİ İŞLEMSEL SÜREÇ BECERİSİ**

Bilgi işlemsel süreç becerileri, belirlenen probleme yönelik çözüm bulma uğraşdır. Bu sebeple bu beceri bilimsel süreç becerilerinden ayrı düşünülemez. Bu becerinin kazanılması sürecinde öğrenciler bir problemi çözmeye çalışırken bilimsel süreç becerilerini kullanmakta, kodlama, programlama, veri toplama, işleme ve dönüştürme gibi becerileri kazanmaktadır. Tasarımsal ve teknik becerilerin geliştirilmesi süreçleri, algoritmik düşünme, özgün düşünme ve inovasyon, problem çözme ve eleştirel düşünme becerilerinin gelişmesine katkı sağlayacaktır.

- Soyutlama Becerisi
- Algoritmik Düşünme Becerisi

- Kodlama, Programlama Becerisi
- Veri Toplama, İşleme, Dönüştürme Becerisi

### 2.2.4. ÜST BİLİŞSEL SÜREÇ BECERİSİ

Üst bilişsel süreç becerileri ile öğrenciler bir sistem tasarlama, sistemin inşası, testlerin yapılması, hataların ayıklanması, sistemin geliştirilmesi gibi önemli kazanımları elde ederler. Bu aşamada öğrenciler hayal ettikleri projeleri tasarlamadan düzenlenmiş örneğini sunmaya kadar üst düzey becerilere ulaşabilirler.

- Sistem Tasarlama Becerisi
- Sistem Kurma (İnşaa Etme) Becerisi
- Test Etme ve Hata Ayıklama Becerisi
- Teknik İşlem Becerisi

### 2.2.5. VERİ TANIMA, İŞLEME, ANALİZ BECERİSİ

Bilişim teknolojileri ve yazılım dersi, öğrencilere temel veri tanıma, işleme ve analiz becerilerini kazandırma amacını taşır. Verinin tanınması, veri kaynaklarını belirleme ve gerekli verileri ayıklama yeteneğini içerir. Verinin işlenmesi, bu verileri düzenleme, filtreleme ve dönüştürme becerisini kapsar. Analiz becerileri ise öğrencilere veriyi anlama, örüntüleri keşfetme ve anlamlı sonuçlar çıkarma yeteneğini kazandırır. Bu beceriler, öğrencilere bilgiye dayalı kararlar verme, problemleri çözme ve iş dünyasında veri odaklı projelerde etkili bir şekilde çalışma yeteneğini sağlar.

- Veri Toplama
- Veri Gösterimi ve Analizi
- Model Doğrulama
- Test ve Doğrulama
- Algoritma ve Süreçler
- Kontrol Yapıları

### 2.2.6. 21. YÜZYIL BECERİLERİ

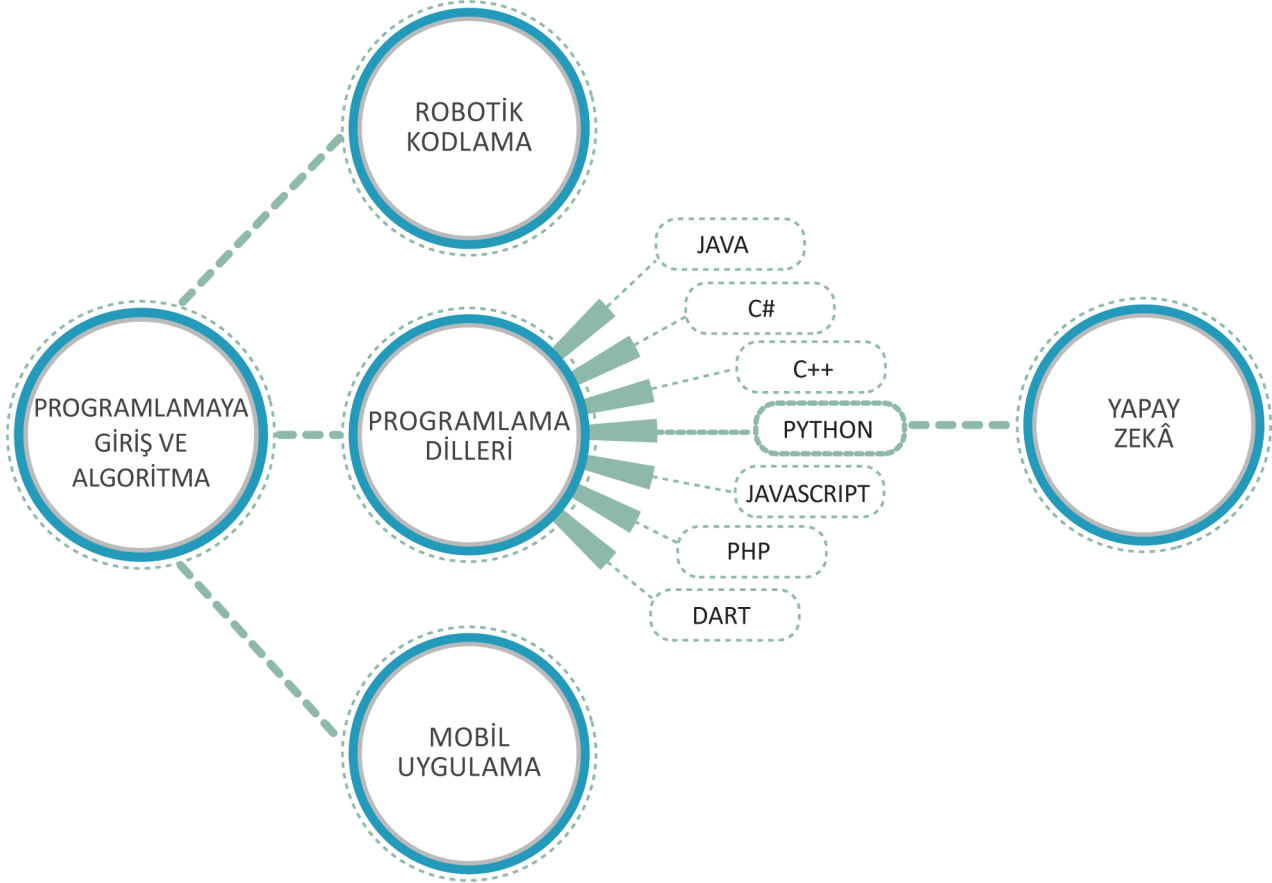
Öğrencilerin 21. yüzyılın hızla değişen iş ve teknoloji ortamında başarılı olabilmelerini sağlayacak çeşitli becerileri kazanması bilişim teknolojileri ve yazılım dersinin amaçlarından biridir. 21. yüzyıl becerileri arasında bilgisayar programlama ve donanım konularının yanı sıra eleştirel düşünme, problem çözme, iş birliği, iletişim, yaratıcılık ve dijital yurttaşlık gibi yetenekler bulunmaktadır. Öğrenciler, karmaşık problemleri analiz etme ve çözme yeteneklerini kazanırken dijital araçlarla etkili bir şekilde iletişim kurmayı öğrenirler. Aynı zamanda öğrencilere teknolojinin etik kullanımı, güvenlik ve gizlilik konularında bilinçli olma yetilerini de kazandırmak hedeflenir.

## 2.3. ÖĞRETİM PROGRAMININ UYGULANMASINDA DİKKAT EDİLECEK HUSUSLAR

Öğrenciler “İnsan, Toplum ve Bilim” alanında yer alan “bilişim teknolojileri ve yazılım” dersinde oluşturulan modüllerden; programlamaya giriş ve algoritma, robotik kodlama, programlama dilleri, mobil uygulama geliştirme, yapay zekâ uygulamaları modüllerinden birini seçerler.

Programlama dilleri modülünde; Java, C++, C#, Dart, PHP, Javascript, Python programlama dillerinden birisi seçilir.

Programlamaya giriş ve algoritma modülü alınmadan; robotik kodlama, mobil uygulama geliştirme ve programlama dilleri modülleri seçilemez. Programlama dilleri modülü alınmadan yapay zekâ uygulamaları modülü seçilemez.



Şema 2 : Bilişim Teknolojileri ve Yazılım Dersi Modül Önceliklendirmesi

Öğrencilerin problemin ne olduğunu anlamaları, günlük hayatta karşılaşılan problemlere örnekler vermeleri, bu problemlerin hangi çözüm yolları denenerek çözülebileceği bu modül kapsamında sınıf ortamında paylaşılır. Bilişim teknolojilerinin problem çözmedeki yeri irdelenir.

Problem çözme sürecinde öncelikle problemleri bileşenlere ayırmak gerektiği ve problem çözme sürecinin döngüsel yapısı anlatılır.

Bilişim teknolojilerinin problem çözme prensipleri, problem çözerken bilişimsel yaklaşımların güçlü ve zayıf yönleri irdelenir. Ayrıca yazılımların günümüz dünyasında sürekli değişen problemlere ayak uydurması için ne tür iyileştirmeler yapıldığından bahsedilir.

Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı'nı uygulamadan önce programın genel amaçları ve kapsamı, derse özgü beceriler, öğrenme-öğretme süreci, ölçme ve değerlendirme, programın uygulanmasına ilişkin esaslar ve programın yapısı bölümleri incelenmeli, öğretim programının felsefesi, öğrenme-öğretme süreci ile ölçme ve değerlendirme yaklaşımları özümsemelidir. Öğrenme ve öğretme süreci işletilirken program

felsefesine uygun hareket edilmelidir. Bu bağlamda aşağıda verilen hususlara dikkat edilmesi gerekmektedir:

- Öğrenme hedeflerinin veriliş düzeni, amaçlı bir sırayı yansıttığı için uygulamada bu sıraya uyulması önemlidir.
- Öğretim programı uygulanırken öğrencilerin tutum ve değerleri kazanmasına özen gösterilmelidir. Tutum ve değerler konuya entegre edilerek verilmelidir.
- Öğretim programı uygulanırken alan becerileri ve genel beceriler konuya ve etkinliklere entegre edilerek verilmelidir.
- Öğretim programı uygulanırken mümkün olduğunca teknolojinin de kullanılabileceği ortamlar oluşturulmalıdır.
- Öğrencilerin planlanan süreden önce hedeflerini tamamlamaları durumunda konuya yönelik çeşitli ek öğrenme faaliyetlerine yer verilmelidir.
- Öğrenme hedeflerinin dışında öğrencilerin ilgilerine bağlı olarak ek bilgi kazandırılmasına ve becerilerinin geliştirilmesine yönelik alıştırmalar yaptırılabilir.
- Öğretim programında yer alan konuların aktarılmasında, diğer disiplinlerle ve günlük hayatla ilişkilendirilmesine imkân sağlayan etkinliklere ve çalışmalara yer verilmelidir.
- Konular işlenirken öğrenme hedefleri ve göstergeleri çerçevesinde öğrencilerin çok boyutlu düşünebilme becerisinin geliştirilmesine yönelik etkinliklere yer verilmelidir.
- Öğretim programı uygulanırken ders saati dışında konu ile ilgili gözlem ve inceleme çalışması yapılması teşvik edilmeli, bu konuda öğrencilere rehberlik yapılmalıdır.

## 2.4. ÜNİTE, KAZANIM SAYISI VE SÜRE TABLOSU

MODÜL	ÜNİTE	KAZANIM SAYISI	DERS SAATİ	YÜZDE ORANI (%)
PROGRAMLAMAYA GİRİŞ VE ALGORİTMA	1. Ünite: Bilişim Teknolojilerinin Programlamadaki Yeri	7	8	13
	2. Ünite: Algoritma ile Problem Çözme ve Akış Diyagramı	11	20	25
	3. Ünite: Programlamanın Temel Kavramları	6	14	20
	4. Ünite: Algoritma ve Akış diyagramını Test Etme	4	14	20
	5. Ünite: İleri Düzey Algoritma Uygulamaları	2	16	22
ROBOTİK KODLAMA	1. Ünite: Robotik Sistemler Ve Tarihsel Gelişimi	8	18	25
	2. Ünite: Robotik Bileşenler	11	25	25
	3. Ünite: Proje Geliştirme	5	18	50
MOBİL UYGULAMA GELİŞTİRME	1. Ünite: Mobil Programlama Okuryazarlığı	9	18	25
	2. Ünite: Mobil Uygulama Tasarımı ve Arayüzü	9	18	25
	3. Ünite: Program Bileşenleri ile Çalışma	8	36	50
PROGRAMLAMA DİLLERİ (JAVA)	1. Ünite: Java Programlamaya Giriş	2	9	13
	2. Ünite: İlkel Veri Tipleri	4	9	13
	3. Ünite: Sınıflar ve Nesneler	3	9	13
	4. Ünite: Karşılaştırma Operatörleri, Mantıksal Operatörler ve Koşullu İfadeler	2	9	13
	5. Ünite: Döngüler	3	9	12
	6. Ünite: Array ve ArrayList Sınıfı	5	9	12
	7. Ünite: Kalıtım	2	9	12
	8. Ünite: Recursive Fonksiyon	2	9	12
PROGRAMLAMA DİLLERİ (C++)	1. Ünite: C++ Çalışma Ortamı	3	14	20
	2. Ünite: Karar ve Döngü Yapıları	3	14	20
	3. Ünite: Fonksiyonlar ve Diziler	4	16	20
	4. Ünite: Formlar	3	14	20
	5. Ünite: Veri Tabanı İşlemleri	6	14	20
PROGRAMLAMA DİLLERİ (C#)	1. Ünite: C# Çalışma Ortamı	5	12	18
	2. Ünite: Karar ve Döngü Yapıları	3	12	16
	3. Ünite: Sınıflar	7	12	18
	4. Ünite: Diziler ve Koleksiyonlar	3	12	16
	5. Ünite: Formlar	4	12	16
	6. Ünite: Veri Tabanı İşlemleri	6	12	16
PROGRAMLAMA DİLLERİ (DART)	1. Ünite: Dart Temelleri	8	18	25
	2. Ünite: Nesne Tabanlı Programlama ve Asenkron Programlama	9	18	25
	3. Ünite: Derinlemesine Dart	6	36	50

PROGRAMLAMA DİLLERİ (PHP)	1. Ünite: PHP Temel Kavramlar	4	18	25
	2. Ünite: PHP'yi Tanıyalım	3	18	25
	3. Ünite: PHP Fonksiyonları, Aritmetiksel ve Mantıksal işlemler	3	18	25
	4. Ünite: PHP ile Veri Tabanı İşlemleri	3	18	25
PROGRAMLAMA DİLLERİ (JAVASCRIPT)	1. Ünite: Javascript Temelleri	8	18	25
	2. Ünite: Javascript Fonksiyonları ve Nesne Tabanlı Programlama	11	18	25
	3. Ünite: Javascript ile Web Geliştirme	14	36	50
PROGRAMLAMA DİLLERİ (PYTHON)	1. Ünite: Python Programlamaya Giriş	7	18	25
	2. Ünite: Python Programlama Dili ile Yazılım Geliştirme	5	18	25
	3. Ünite: Nesne Yönelimli Proje Geliştirme	9	36	50
YAPAY ZEKÂ UYGULAMALARI	1. Ünite: Veri Bilimi Temel Kavramları	16	16	22
	2. Ünite: Temel İstatistik	3	7	10
	3. Ünite: Değişim ve Veri	2	7	10
	4. Ünite: Veri Görselleştirme	5	7	10
	5. Ünite: Veriler Arası İlişki	3	7	10
	6. Ünite: Veri Manipülasyonu	7	7	9
	7. Ünite: Veri Uygulamaları	2	7	9
	8. Ünite: Yapay Zekâya Giriş	4	7	10
	9. Ünite: Yapay Zekâ Tabanlı Oyun Geliştirme	3	7	10

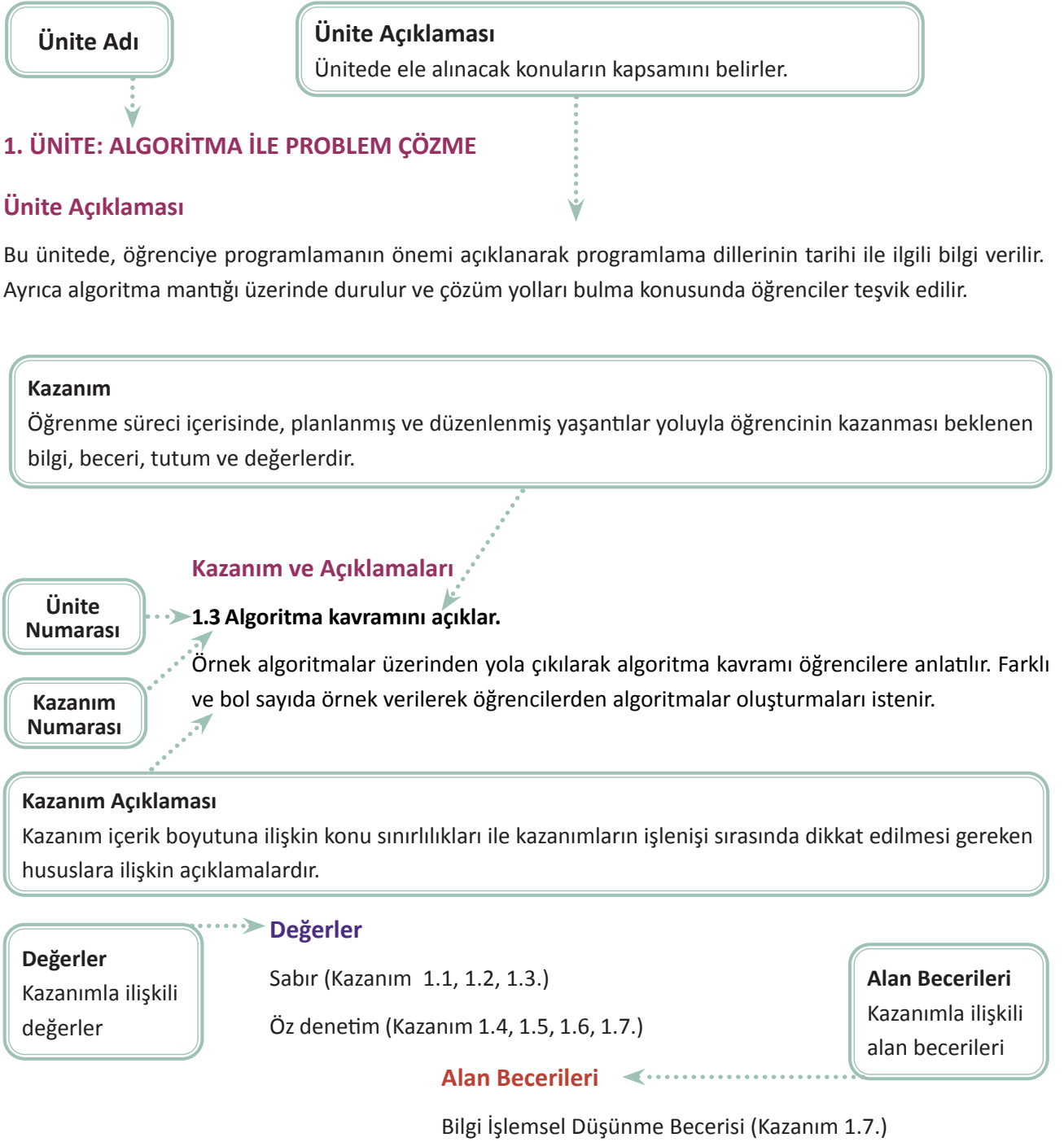
## 2.5. KİTAP FORMA SAYILARI VE KİTAP EBADI

DERS KİTABI	FORMA SAYILARI	KİTAP EBADI
PROGRAMLAMAYA GİRİŞ VE ALGORİTMA	13-14	19,5 x 27,5
ROBOTİK KODLAMA	19-20	19,5 x 27,5
MOBİL UYGULAMA GELİŞTİRME	19-20	19,5 x 27,5
PROGRAMLAMA DİLLERİ (JAVA)	20-21	19,5 x 27,5
PROGRAMLAMA DİLLERİ (C++)	13-14	19,5 x 27,5
PROGRAMLAMA DİLLERİ (C#)	17-18	19,5 x 27,5
PROGRAMLAMA DİLLERİ (DART)	13-14	19,5 x 27,5
PROGRAMLAMA DİLLERİ (PHP)	13-14	19,5 x 27,5
PROGRAMLAMA DİLLERİ (JAVASCRIPT)	18-19	19,5 x 27,5
PROGRAMLAMA DİLLERİ (PYTHON)	18-19	19,5 x 27,5
YAPAY ZEKÂ UYGULAMALARI	21-22	19,5 x 27,5



### 3.1. ÖĞRETİM PROGRAMININ YAPISI

Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı'nın ünite yapısı şematik olarak aşağıda sunulmuştur:



## 3.2. ÜNİTE, KAZANIM VE AÇIKLAMALAR

### PROGRAMLAMAYA GİRİŞ VE ALGORİTMA

Bilgi toplumu içerisinde hayatımızın her alanında karşılaşılan bilgisayar programlarının hazırlanması sürecine programlama adı verilir. Bu süreçte algoritmalar büyük bir önem arz etmektedir. Programlama sürecinin adımlarını planlarken ilk aşama programın algoritmasının oluşturulmasıdır. Bilgisayar bilimlerinde algoritma, bilgisayara bir problemi nasıl çözeceğini veya bir görevi nasıl tamamlayacağını söyleyen kesin talimatlar dizisidir.

Tüm programlama dillerinin temeli algoritmaya dayanır. Algoritma mantığı ve bilgi işlemsel düşünme ile problem çözme becerileri 21. yüzyılın temel becerilerinden kabul edilmektedir. Hızlı gelişen dijital dünyada bilgisayar programlarını anlama ve geliştirme bilgisi de bireylerin sahip olması gereken önemli beceriler arasındadır. Yaşam boyunca çözülmesi gereken pek çok problemle karşı karşıya kalınmaktadır. Problemleri çözerken çözüme giden farklı yollar bulunur. Bu farklı yollardan en uygununu seçmek bireylerin karar verme yeteneklerini geliştirir. Algoritma ve programlama bilgisi, öğrencilerin analitik ve yaratıcı düşünebilmelerini sağlar. Programlama bilgisi birçok farklı kariyer fırsatı sunar.

Bu ders ile öğrencilerin;

- Programlama dilleri ve algoritma kavramları ve gelişim süreci hakkında bilgi sahibi olmaları,
- Bir problemin çözüm aşamalarını oluşturmaları ve bunu bir algoritma şeklinde ortaya koyabilmeleri,
- Bir problem için akış diyagramı oluşturabilmeleri,
- Tasarladıkları algoritmayı bir programlama dili ile kodlayabilecekleri bilgisini kazanmaları,
- Programlamanın temel kavramlarını öğrenmeleri,
- Döngü ve karar yapıları içeren program geliştirebilmeleri,
- Tasarlanan algoritma ve akış diyagramlarındaki hataları listeleyebilmeleri ve düzeltebilmeleri,
- İleri düzey algoritma uygulamalarını ve kullanım yerlerini açıklayabilmeleri amaçlanmaktadır.

## 1. ÜNİTE: BİLİŞİM TEKNOLOJİLERİNİN PROGRAMLAMADAKİ YERİ

### Ünite Açıklaması

Bilişim teknolojileri, programlama dünyasının temel yapılarından biridir. Programlama, bilgisayarlar ve diğer dijital cihazlar tarafından anlaşılabilir komutlar dizisi olarak yazılır ve bu komutlar, bilişim teknolojileri kullanılarak işlenir. Bilişim teknolojileri yazılım geliştirme, veri tabanı yönetimi, ağ iletişimi, web uygulamaları, mobil uygulamalar gibi birçok alanda kullanılır.

Bilişim teknolojileri, programlamayı kolaylaştırır ve program geliştiricilere entegre geliştirme ortamları (IDE'ler), veri tabanları ve diğer yazılım geliştirme araçları gibi güçlü araçlar sunar. Bilgi işleme, depolama ve iletişim konularında da programlara destek sağlar.

### Kazanım ve Açıklamaları

#### 1.1. Bilişim teknolojilerine ilişkin temel kavramları tanımlar.

Bilgi, teknoloji ve iletişim kavramlarının açıklaması yapılır.

**1.2. Bilişim teknolojisi araçlarını listeler.**

- a) Ülkemizde teknolojik alanda kullanılan araçlar örneklendirilir. Örneğin Türk uzay yolcusu ve bilim misyonu gibi bilimsel projeler tanıtılır.
- b) Bu araçların insanların hayatlarını kolaylaştırmak ve ihtiyaçlarını gidermek için kullanıldığı, kullanım alanlarının esnek olduğu vurgulanır.

**1.3. Bilişim teknolojileri cihazlarının gelişimini açıklar.**

- a) Ülkemizde yaygın olarak kullanılan bilişim teknolojisi cihazlarının (bilgisayar, telefon, tablet, hesap makinesi, vb.) tarihsel gelişimi ile ilgili bilgilendirme yapılır. Örneğin günümüzde üretilen yerli otomobilin mekanik sistem araçlarından bahsedilir.
- b) Gelecekte ne gibi gelişmeler olabileceği hakkında fikir yürütülür.
- c) Bilişim cihazlarının gelişimini takip etmenin kendi geleceği ve meslek hayatı için sorumluluğu olduğunu bilir.

**1.4. Bilişim teknolojilerinin gelişiminde önemli rol oynayan keşifler, buluşlar, olaylar ve dönüm noktalarını tartışır.**

- a) Bilişim teknolojisi araçlarının keşfinden günümüze olan gelişimi örneklendirilir. Tarihte Cezeri'nin şifreli kilit sisteminin, günümüz kriptoloji biliminin temelini oluşturduğu açıklanır.
- b) Teknolojik araçların gelişimine yön veren kişiler ve bu araçların insanlık tarihinde yarattığı dönüm noktaları tartışılır. Telefonun icat edilmesiyle birlikte iletişim araçlarındaki değişimin, zamanla mobil uygulama geliştirme gerekliliğini ortaya çıkardığından bahsedilir .

**1.5. Bilişim teknolojilerinin kullanım alanlarını açıklar.**

- a) Bilişim teknolojilerinin günümüzde kendine birçok alanda yer bulduğundan bahsedilir. Kullanım alanlarına dair tahmin yürütülmesi istenir.
- b) Bilişim teknolojilerinin eğitim, sağlık, iletişim, güvenlik, ulaşım gibi birçok alanda kullanıldığı anlatılır ve tüm bu alanlara örnekler verilerek konu pekiştirilir.

**1.6. Bilişim teknolojilerinin olumlu ve olumsuz yönlerini tartışır.**

- a) Teknolojik gelişmelerin olumlu yönleri olduğu kadar olumsuz yönlerinin de olabileceği vurgusu yapılır. Olumlu ve olumsuz yönlerin neler olabileceği tartışılır.
- b) Olumlu ve olumsuz yönlere örnekler verilerek konu kavratılır.
- c) Gelecekte olumsuz sonuçlar doğurabilecek teknolojilerin varlığı tartışılır.

**1.7. Teknolojik gelişmeleri takip etmenin önemini farkına varır.**

- a) Teknolojinin sürekli gelişen ve yenilenen bir kavram olduğu vurgulanır. Bu yenilikleri takip etmenin ülkesi için önemini kavrar.
- b) Teknolojik yenilikleri takip eden bir birey olmasının ülkesine karşı sorumluluklarından biri olduğunu bilir.
- c) Öğrenciler bilişim teknolojileri alanındaki son gelişmelere yönelik (yapay zekâ vb.) araştırma yapmaya yönlendirilir.

- ç) Gelişmeleri takip edebilmek, alanda güncel kalabilmek için yaşam boyu öğrenme kavramından bahsedilir, yaşam boyu öğrenmenin önemi vurgulanır.

### Değerler

Özdenetim (Kazanım 1.1., 1.3., 1.5., 1.6.)

Sorumluluk (Kazanım 1.2., 1.3.)

Vatanseverlik (Kazanım 1.1, 1.4., 1.7.)

### Alan Becerileri

İletişim Becerisi (Kazanım 1.1, 1.2, 1,4)

Soyutlama Becerisi (Kazanım 1.2, 1.5, 1.7)

Tahmin Etme Becerisi (Kazanım 1.3, 1.6)

İşbirlikli Çalışma Becerisi (Kazanım 1.3, 1.4)

## 2. ÜNİTE: ALGORİTMA İLE PROBLEM ÇÖZME VE AKIŞ DİYAGRAMI

### Ünite Açıklaması

Akış diyagramları, bir sürecin işlem adımlarını sıralı bir şekilde özetlemek için semboller kullanan görsel şemalardır. Akış diyagramları programcıların en verimli kodlamayı geliştirmelerine ve programcıların potansiyel sorunlu alanın nerede olduğunu anlamalarına yardımcı olur.

Bu üniteye sırasıyla; programlamanın önemi, programlama dillerinin gelişimi, algoritma kavramı, çözümü istenen bir problemin kavranması, gereksinimleri ve çözümü için gerekli olan en doğru algoritmanın yazımı, akış diyagramının nerelerde kullanıldığı, akış diyagramı şekilleri ve elemanlarının listesi, var olan akış diyagramından problemin kavranması ve bir problemin akış diyagramının tasarlanması üzerinde durulur.

### Kazanım ve Açıklamaları

#### 2.1. Çözümü istenen problemi kavrar.

- Algoritma çözüm sürecinde en önemli aşamanın problemi anlamak ve problemin çözümüne yönelik adımları tahmin etmek olduğu açıklanır. Çözüm gerektirecek ortak problemler tartışılır.
- Probleme neden olan durumların araştırılması gerektiği belirtilir.
- Problemin etki alanı sorularla buldurmaya çalışılır.
- Çözüme ulaşmış benzer problem örnekleri verilerek çözüm hakkında bilgi sahibi olunması sağlanır.
- Problemde verilen ve istenen bilgileri listelemenin, problemi daha iyi anlayabilmek için önemli olduğu vurgulanır.
- Eğer verilen problem karmaşık bir problemse problemin alt basamaklara ayrılacağı belirtilir.

#### 2.2. Çözüm için gereksinimlerini belirler.

- Belirlenen problem durumunun çözümünü kolaylaştıracak unsurlar üzerinde durulur. Problemin açık

anlaşılır ve net hâle getirilmesi sağlanır.

- b) Günlük hayatta karşılaşılan problemlere çözüm önerileri sunulur.
- c) Bir problemin çözümü için farklı çözüm yolları olabileceği vurgulanır.
- ç) Problem çözme süreci ile ilgili geliştirilmiş kuram ve yaklaşımlardan bahsedilerek bu sürecin temel basamakları açıklanır.
- d) Bir problemin çözümü için kullanılacak temel yöntem ve teknikler ile ilgili öğrencilere bilgilendirme yapılır.

### **2.3. Problemin girdi, çıktı ve işlem aşamalarını belirler.**

- a) Öğrencilere girdi ve çıktı kavramları açıklanır.
- b) Örnek problem durumlarının girdi ve çıktıları belirlenir.
- c) Problem alt basamaklara ayrılarak çözüm için işlem basamakları belirlenir.
- ç) İşlem aşamaları belirlenirken bulunan farklı çözüm yollarından çözüme en kısa sürede ulaştıracak olan seçilmeye çalışılır.

### **2.4. Algoritma kavramını açıklar.**

- a) Günlük hayattaki rutin işler üzerine öğrencilerin fikri alınır. Bu işlerin mantığı ve sırası değiştiğinde işin sonucuna nasıl etki edeceğini söyler.
- b) Örnek algoritmalar üzerinden yola çıkılarak algoritma kavramı öğrencilere anlatılır.
- c) Algoritma kelimesinin 9. yüzyılda yaşamış matematikçi Ebu Cafer Muhammed bin Musa el Harezmi'nin adının Latince okunuşundan geldiği açıklanır.
- ç) İlk algoritmanın, el-Harezmi tarafından "Hisab el-cebir ve el-mukabala" kitabında sunulduğundan bahsedilir.
- d) Öğrencilerden el-Harezmi ve onun algoritma ve bilgisayar bilimine katkılarıyla ilgili araştırma yapmaları istenir. Araştırma sonuçları sınıfla paylaşılır.

### **2.5. Bir problemin çözümü için en doğru algoritmayı geliştirir.**

- a) Belirlenen girdi ve çıktıları kullanarak işlem adımlarını doğru sıralamanın önemi açıklanır.
- b) Algoritma yazımında oluşabilecek hatalar üzerinde düşünülmesi istenir.
- c) Toplumu ilgilendiren bir problemin çözümüne yönelik öğrencilerden girdi, çıktı ve işlem adımlarıyla algoritma yazmaları istenir.
- ç) Tasarlanan algoritmanın doğruluğu tartışılır. Geliştirilmesi gereken noktalar yeniden çalışılarak algoritma en doğru ve işlevsel hâle getirilir.

### **2.6. Programlamanın önemini yorumlar.**

Öğrencilere programlama kavramının ne olduğu açıklanır. Neden önemli olduğu saygı çerçevesinde tartışılır.

### **2.7. Programlama dillerinin gelişimini açıklar.**

- a) Programlama dillerinin farklı kullanım amaçları ile ilgili bilgilendirme yapılır.

b) Programın amacına yönelik programlama dili kullanılabileceği açıklanır.

**2.8. Akış diyagramının kullanım amaçlarını açıklar.**

- a) Akış diyagramı kavramı öğrencilere açıklanır.
- b) Karmaşık bir problemin çözümüne yönelik akış diyagramı oluşturmanın sağlayacağı avantajlar üzerinde durulur.
- c) Akış diyagramı tasarlarken herkes tarafından anlaşılır olabilmesi için standart anlamlar taşıyan şekillerin kullanılması gerektiği anlatılır.
- ç) Akış diyagramlarının programın akışını görebilmek ve süreçteki hatayı bulmakta kolaylık sağladığı vurgulanır.

**2.9. Temel akış diyagramı şekillerini ve elemanlarını listeler.**

- a) Temel akış diyagramı şekilleri ve yardımcı elemanları öğrencilere gösterilir. Şekillerin anlamları açıklanır, kullanıldığı durumlar örneklendirilir.
- b) Akış diyagramının tüm sembollerinin oklarla bağlı olması ve diyagramın yönünün belli bir yönde hazırlanması gerektiği vurgulanmalıdır.
- c) Akış diyagramlarının içerik ve türlerine göre; doğrusal, mantıksal ve döngü içeren akış diyagramları olmak üzere üçe ayrıldığı söylenir. Daha iyi anlaşılması için örnek akış diyagramları incelenir.

**2.10. Var olan bir akış diyagramını okuyarak problemi kavrar.**

- a) Akış diyagramının girdi ve çıktıları belirlenir. Girdi ve çıktılardan yola çıkarak akış diyagramının hangi amaca hizmet edebileceği tahmin edilir.
- b) Akış diyagramının türünü belirlerken öncelikle diyagram içerisinde karar veya döngü yapıları aranması gerektiği vurgulanır.
- c) Diyagramda istenen veri girişleri tespit edilerek yapılacak işlemler listelenir ve istenilen sonuç bulunur.

**2.11. Problemin akış diyagramını tasarlar.**

- a) Belirlenen problem durumuna yönelik akış diyagramı türü seçildikten sonra uygun semboller kullanılarak diyagram taslak olarak çizilir.
- b) Akış diyagramının elektronik ortamdaki çizimi için kelime işlemci programları veya diğer çizim programları kullanılır.
- c) Günlük hayatta yaygın olarak kullanılan bir uygulamada akış diyagramı tasarlanır.

**Değerler**

Sabır (Kazanım 2.1., 2.2., 2.3., 2.8., 2.9., 2.10., 2.11.)

Öz denetim (Kazanım 2.4., 2.5., 2.6., 2.7., 2.8., 2.9.)

Sorumluluk (Kazanım 2.2., 2.6., 2.7., 2.9., 2.10., 2.11.)

Yardımselverlik (Kazanım 2.6., 2.7., 2.11.)

Saygı (Kazanım 2.6, 2.8., 2.10., 2.11.)

**Alan Becerileri**

Soyutlama Becerisi (2.1., 2.2., 2.3., 2.4., 2.8., 2.9., 2.10., 2.11.)

Mantıksal Düşünme Becerisi (2.1., 2.2., 2.3., 2.4., 2.8., 2.9., 2.10., 2.11.)

Test Etme ve Hata Ayıklama Becerisi (Kazanım 2.4., 2.5., 2.6., 2.7.)

Algoritmik Düşünme Becerisi (Kazanım 2.7., 2.10., 2.11.)

İş birlikli Çalışma (Kazanım 2.10.)

**3. ÜNİTE: PROGRAMLAMANIN TEMEL KAVRAMLARI****Ünite Açıklaması**

Bu üniteye sırasıyla; programlamada kullanılan işlem ve semboller, veri türleri, karar ve döngü yapıları ve ikisinin birlikte kullanıldığı bir program tasarlama üzerinde durulur.

**Kazanım ve Açıklamaları****3.1. Programlamada kullanılan işlem ve sembollerini listeler.**

- Programlama dillerinde basit farklılıklar gözlemlense de ortak kullanılan kavramlar olduğu açıklanır. Operatörler, değişkenler, koşullar, döngüler gibi temel programlama kavramlarının gerekliliği ve programın akışını yönlendirmesi üzerinde durulur.
- Matematikte kullanılan sembollerle programlamada kullanılan sembollerin benzer ve farklı yönleri gösterilerek kullanım amaçları açıklanır.
- Yazılan programlama diline göre aynı sembollerin farklı anlamlar ifade edebileceği söylenir, bu durum örneklendirilir.

**3.2. Veri türlerini amacına uygun şekilde programa tanımlar.**

- Veri kavramının tanımı yapılarak farklı veri türleri olduğu ifade edilir.
- Veri türlerini doğru bir şekilde tanımlamanın programlamadaki önemi üzerinde durulur.
- Farklı programlama dillerine ait veri türleri olduğu dile getirilir. Programda değişkenlerde tutulacak verinin kullanım amacına uygun olarak saklanmasının önemi kavranır. Aynı verinin farklı amaçlarda kullanılabileceği ve programa tanımlanabileceği durumlar örneklendirilir.
- Kullanılan programlama diline göre veri türlerinin isimlendirmesinin değişebildiğinden bahsedilir.
- Sayısal veya metinsel veri türü tanımlamaları örnekler üzerinden gösterilir.

**3.3. Farklı veri türlerini program içerisinde kullanır.**

- Her program farklı bir amaca hizmet edeceğinden ihtiyaç duyabileceği veri türlerinin farklı olabileceği açıklanır. Bu veri türlerinin programın amacına yönelik tanımlanması sağlanır. Tanımlanan verilerin sayı, metin, dizi gibi ait olduğu programın veri türüyle kullanılması sağlanır.
- Seçilen veri türüne göre alt veri türleri açıklanır.

- c) Sayısal veri türünde tam sayı veya ondalıklı sayı kullanımında tanımın değişeceği, program içerisinde veri türüne göre sonuçların etkilenebileceği örneklendirilir.
- ç) Byte, integer, long, float, string, boolean, char kavramları örneklerle açıklanır.
- d) Birden fazla aynı tür veriyi depolayabilen dizi kavramından bahsedilir.

**3.4. Karar yapılarını kullanarak programı geliştirir.**

- a) Farklı problem durumlarında programın akışına yön vermenin gerekliliği açıklanır.
- b) Karar yapılarında önemli bir yeri olan mantıksal operatörler ile ilgili bilgi verilir.
- c) Programlama dillerinde kullanılan if-else ve else if yapılarına örnekler verilir.
- ç) Öğrencilerin birden fazla koşul durumuyla program tasarlaması sağlanır.

**3.5. Döngü yapılarını kullanarak programı geliştirir.**

- a) Programda tekrar eden yapıları kullanmanın sağladığı kolaylıklar anlatılır. Programcının iş yükünü hafifletecek farklı döngü yapıları olduğundan bahsedilir.
- b) Programlamada döngü yapılarını kullanmanın kod tekrarının önüne geçerek daha anlaşılır ve sade programlar yazmak için önemli olduğu vurgulanır.
- c) Hangi döngü yapısının kullanılması gerektiğine programın amacına yönelik karar verileceği belirtilir. For, while vb gibi temel döngü yapılarının işlevi açıklanır.
- ç) Döngünün temel taşları olan başlangıç ve bitiş değeri, artış miktarı ve devam etme koşulunun ne olduğu ifadeleri açıklanır.
- d) Verilen bir problemi döngü yapısı kullanarak ve kullanmadan çözmeleri istenerek aradaki farkı görmeleri sağlanır.

**3.6. Döngü ve karar yapıları ile program tasarlar.**

Öğrencilere döngü ve karar yapılarını içinde barındıran problemler verilerek bu problemin algoritmasını tasarlamaları ve akış diyagramlarını çizmeleri istenir.

**Değerler**

Adalet (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)

Dürüstlük (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)

Öz denetim (Kazanım 3.3., 3.4., 3.5., 3.6.)

Saygı (Kazanım 3.4., 3.5., 3.6.)

Sabır (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)

**Alan Becerileri**

Soyutlama Becerisi (Kazanım 3.1., 3.2., 3.3.)

Algoritmik Düşünme Becerisi (Kazanım 3.2., 3.3., 3.4., 3.5., 3.6.)

Kodlama, Programlama Becerisi (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)



## 4. ÜNİTE: ALGORİTMA VE AKIŞ DİYAGRAMINI TEST ETME

### Ünite Açıklaması

Hata ayıklama, bir problemin çözümü için tasarlanan algoritmanın düzensiz davranmasına veya çalışmamasına neden olabilecek hataları bulma ve düzeltme işlemidir. Programlamada hata ayıklama, yazılan bir programın düzgün çalışması açısından önemli rol oynar.

Bu üniteye sırasıyla; tasarlanan algoritma ve akış diyagramının test edilmesi, hatalarının listelenmesi ve hataların düzeltilerek algoritmanın iyileştirilmesi üzerinde durulur.

### Kazanım ve Açıklamaları

#### 4.1. Tasarlanan algoritma ve akış diyagramını test eder.

- Öğrencilere tasarladığı algoritma ve akış diyagramlarının, kodlamaya geçmeden mantıksal olarak incelenmesi, olası sonuçlar düşünülerek her ihtimalin göz önünde bulundurulması gerektiği, algoritma ve akış diyagramının her durumda doğru sonuçlar üretip üretmediğinin test edilmesi gerektiği vurgulanır.
- Algoritmaları akış diyagramları üzerinden test etmek için kullanılan çeşitli araçlardan bahsedilir.
- Basit bir algoritmanın akış diyagramı çizilerek test edilmesi sağlanır.
- Programlamada algoritmanın test edilmesinin doğru programlar yazmadaki önemi vurgulanır.

#### 4.2. Tasarladığı algoritma ve akış diyagramının hatalarını listeler.

- Hata ayıklama ve hata yakalama yöntemlerinden bahsedilir.
- Hatanın türü belirlenmeye çalışılır. Hatanın algoritmada mı akış diyagramında mı olduğu tespit edilir.
- Algoritma ve akış diyagramlarındaki hatalar bulunarak istenilen sonuca ulaşmayı engelleyen tüm hatalar listelenir.

#### 4.3. Hatalı algoritma ve akış diyagramlarının, programın istenilen çıktıyı vermesine engel olacağını fark eder.

- Bir problemin çözümü için yazılan algoritmanın ve tasarlanan akış diyagramlarının hatalı olması durumunda olası sonuçların neler olabileceği tartışılır.
- Programın vermesi istenen çıktısı ile hatalı algoritma sonucu ortaya çıkan çıktılar karşılaştırılarak birlikte değerlendirilir.

#### 4.4. Tasarladığı algoritma ve akış diyagramının hatalarını düzeltir.

- Hatalı bir algoritmanın var olan hatalar listelendikten sonra düzeltilebileceği vurgulanır. Belirlenen hataların nasıl giderileceği üzerine çözüm yöntemleri tartışılır.
- Hatalı algoritma örnekleri saygı çerçevesinde tartışılarak elde edilen bulgular listelenir. Hataların düzeltilerek programın işlevsel hâle getirilmesi sağlanır.
- Algoritma ve akış diyagramlarında karşılaşılan hatalardan yola çıkarak yeni tasarlanan algoritmalarda hata oluşumunu engellemeye yönelik çalışmalar yapılır.
- Hata ayıklaması yapılan algoritmaya eklenilecek farklı girdi ve çıktılar ile program durumu geliştirilir.

**Değerler**

Sabır (Kazanım 4.1., 4.2., 4.3.)

Saygı (Kazanım 4.1., 4.3., 4.4.)

Sevgi (Kazanım 4.1., 4.3.)

Sorumluluk (Kazanım 4.1., 4.2., 4.3.)

Yardımselik (Kazanım 4.1.)

**Alan Becerileri**

Algoritmik Düşünme Becerisi (Kazanım 4.1., 4.2., 4.3.)

Genelleme Becerisi (Kazanım 4.3.)

Test Etme ve Hata Ayıklama Becerisi (Kazanım 4.3.)

İş birlikli Çalışma (Kazanım 4.1., 4.2., 4.3.)

İletişim (Kazanım 4.1., 4.2., 4.3.)

**5. ÜNİTE: İLERİ DÜZEY ALGORİTMA UYGULAMALARI****Ünite Açıklaması**

Bu ünite de sırasıyla; arama ve sıralama algoritmalarının akış diyagramıyla gösterilmesi ve ileri algoritma uygulamalarının ve kullanım yerlerinin açıklanması üzerinde durulur.

**Kazanım ve Açıklamaları****5.1. Arama ve sıralama algoritmalarını akış diyagramıyla gösterir.**

- Algoritmaların farklı tasarlama yöntemleri olduğu açıklanır.
- Belirli durumlara özgü farklı algoritmalar olduğundan bahsedilir. Arama ve sıralama algoritmalarının tanımı yapılarak farkları vurgulanır.
- Doğrusal arama (Linear search) ve ikili arama (Binary search) algoritması kavramlarından bahsedilir. Arama algoritmaları doğrusal arama ve ikili arama olarak ikiye ayrılır ve örnekler verilir. Bu örnekler üzerinden akış diyagramı çizmesi istenir.
- Sıralama algoritmalarının çok basit bir mantıkla küçükten büyüğe sıralama şeklinde başlayıp ne kadar çeşitlenebileceği ve sıralamanın değiştirilebileceği örneklerle açıklanır. Bu örnekler üzerinden akış diyagramı çizmesi istenir.
- Seçerek Sıralama (Selection Sort), Kabarcık Sıralaması (Bubble Sort), Eklemeli Sıralama (Insertion Sort), Birleştirme Sıralaması (Merge Sort), Hızlı Sıralama (Quick Sort) kavramları açıklanır.

**5.2. İleri algoritma uygulamalarını ve kullanım yerlerini açıklar.**

- Çeşitli algoritma tasarlama yöntemleriyle tasarlanan algoritma türleri incelenir. Çalışma prensipleri ve avantaj dezavantajları belirtilen algoritma çeşitlerinin kullanılabileceği program fikirleri üretilir.
- Popüler uygulamalarda kullanılan algoritmaların çalışma mantığı incelenir. Gezgin satıcı probleminde kullanılan algoritma gösterilir.

- c) Veri madenciliği, şifreleme algoritmaları ve yapay sinir ağlarından bahsedilir.
- ç) Veri madenciliğinin sağlık, pazarlama, bankacılık gibi birçok alanda kullanımına örnekler verilerek konunun anlaşılması sağlanır.
- d) Şifreleme algoritmalarının çok eski zamanlardan beri farklı yöntemlerle kullanıldığından bahsedilir. Genellikle verilerin şifrelenmesinde kullanılırken e-izmalarda da yaygın olarak bu algoritmaların kullanıldığından bahsedilir.
- e) Yapay sinir ağları insan beyninin çalışmasından esinlenilerek yapay sinir hücrelerine bilgi aktarabilen sistemler olduğu açıklanır. Yapay zekânın kullanım alanları örneklerle açıklanır.
- f) İleri algoritma uygulamalarına örnekler verilerek (Makine öğrenimi ve yapay zeka, doğal dil işleme, otomasyon gibi) bunların kullanım alanlarının neler olabileceği tartışılır.

### Değerler

Sabır (Kazanım 5.1., 5.2.)

Sorumluluk (Kazanım 5.1., 5.2.)

### Alan Becerileri

Algoritma Ve Süreçler (Kazanım 5.1., 5.2.)

## ROBOTİK KODLAMA

Hızla gelişen teknoloji ile robotik donanımların kullanım alanları da hızla artmaktadır. Bu gelişmeler yüksek işlem kapasitesi, hızlı veri iletimi, veri toplama, çevreyi anlama, hızlı analiz gibi bilgi işleme süreçlerinin hızlanmasını sağlayarak robotiğin fiziki güç gerektiren sektörlerin yanı sıra yaşlı ve çocuk bakımı gibi bireysel destek alanlarında da kullanımının yaygınlaşmasını sağlamaktadır.

Yeni döneme uyum sağlamakta gecikmeyen ülkemizin yapmış olduğu teknolojik hamleler sayesinde insan-sız hava araçları, yarı ve tam otonom sistemler yerli olarak üretilmekte ve güvenliğimiz için çalışmaktadır. Öğrenebilen sistemlerin geliştirilmesi ve işletilmesi çalışmaları dünyada ve ülkemizde başlamış ve geleceğin en önemli teknolojik gelişmeleri arasında gösterilmektedir.

Robot teknolojilerinin hayatın her alanına etki etmeye başladığı günümüzde çocuklarımızı yani yarınlarmızı yalnızca bu teknolojileri kullanan değil aynı zamanda bu teknolojileri üretebilen, kodlayabilen ve geliştirebilen bireyler haline getirmek dersin öncelikli amacıdır.

Robotik Kodlama dersinde öğrencilerin robotiğin gelişimsel tarihini, türlerini ve bileşenlerini öğrenmelerinin yanı sıra, bileşenlerin farklı amaçlarla kullanımını fark edebilmeleri, veri toplama, veri iletimi ve analizi yöntemlerini bilerek kendilerine ait projeler tasarlayabilmeleri ve geliştirebilmeleri hedeflenmektedir.

Robotik Kodlama dersi ile öğrencilerin;

- Algoritma tasarımına ilişkin anlayış geliştirerek sözel ve görsel olarak ifade edebilen,
- Problem çözme becerilerine sahip,
- Öğrendiği bilgileri farklı durumlara aktarabilen,
- Disiplinler arası öğrenme etkinliklerini bilen,
- Robotiğin hayatımızdaki yerine ilişkin bilgilere sahip,
- Günlük hayatta karşılaşılan sorunların (yaşlı ve engelli bireylerin karşılaştığı sorunlar, çevre, ulaşım sorunları vb.) çözümüne ilgili,
- Öğrenme sürecinde iş birlikli çalışma becerisine sahip,
- Robotik ve kodlamanın hayatın her alanında var olduğunu bilen,
- Yaşam boyu öğrenmeye istekli bireyler hâline gelmeleri hedeflenmektedir.

## 1. ÜNİTE: ROBOTİK SİSTEMLER VE TARİHSEL GELİŞİMİ

### Ünite Açıklaması

Bu ünite, robotik ve otomasyonun tarihçesi, temel kavramları, farklı robot platformları, otomatik sistemlerin farklılıkları, öğrenebilen sistemlerin çalışma mantığı anlatılarak bir soruna robotik sistemler ile çözüm getirmeye yönelik süreçler üzerinde durulur.

### Kazanım ve Açıklamaları

#### 1.1. Robotiğin tarihsel gelişimini açıklar.

- a) Robot, otomasyon, sensör ve programlama gibi temel robotik kavramlar ifade edilir.

- b) Robotiğin tarihi, ilk mekanik robotlardan günümüzün yüksek teknoloji robotlarına kadar olan gelişimi anlatılır.
- c) Tarihsel süreçte mekanik olarak gelişen robotik alanına elektronik ve bilgisayarların dahil olmasıyla yaşanan gelişimdeki hızlanmaya dikkat çekilir.
- ç) Gelecekte robot ve insan etkileşiminin ne düzeyde olacağına dair öngörülerde bulunmaları istenir.

**1.2. Robotik kodlamanın endüstriyel ve günlük hayattaki uygulamalarını araştırır.**

- a) Tarihsel süreçte robotiğin kullanım alanlarına, endüstriyel ve günlük hayattaki kullanılan robotların gelişimine ait görsel örnekler verilir.
- b) Günümüzde robotik kullanımının hayatımızı nasıl kolaylaştırdığı ifade edilir.
- c) Üretim ve ürün tasarımında robotik kodlamanın kullanımı incelenir.
- ç) Günlük hayatta karşılaşılan robotik cihazlar (örneğin akıllı ev sistemleri veya otonom süpürgeler) incelenerek bu teknolojinin çalışma mantığına dikkat çekilir.

**1.3. Günümüz sorunlarından birine robotik ile çözüm fikri oluşturur.**

- a) Çevresinde veya dünya genelinde karşılaşılan bir güncel sorunu tanımlaması istenir.
- b) Bu soruna robotikle ilgili herhangi bir çözüm getirilmiş mi araştırması istenir.
- c) Çözüm varsa mevcut çözümü geliştirmesi, yoksa robotik ile çözüm fikri oluşturması istenir.
- ç) Robotun nasıl çalışacağını, hangi sensörlerin kullanılacağını ve hangi verilerin toplanacağını içeren çözüm fikrini sınıf veya topluluk önünde sunması sağlanır.

**1.4. Farklı robot türleri ve mikrodenetleyicileri karşılaştırır.**

- a) Kullanım amaçlarına göre oluşturulan farklı robot türleri (örneğin tekerlekli robotlar, insansız hava araçları, kol robotları) anlatılır.
- b) Aynı robotik mekanizmada çeşitli mikrodenetleyicilerin kullanılabilmesine fakat tüm mikrodenetleyicilerde aynı programlama dilinin kullanılmadığına dikkat çekilir. Bu sebeple seçilecek mikrodenetleyiciye karar verirken kullandığı programlama dilinin de dikkate alınması gerektiği söylenir.
- c) Farklı robot türlerinin avantajları ve dezavantajları karşılaştırılır. Örneğin, tekerlekli robotların karada hareket etme yeteneği ve istikrarı, ancak sınırlı engel aşma kabiliyeti olabilir.
- ç) Farklı mikrodenetleyicilerin avantajları ve dezavantajları karşılaştırılır. Örneğin kodlama dili yazım kolaylığı ve kütüphane desteği, donanım özellikleri, güç kullanımı.

**1.5. Otomatik sistemlerin çalışma mantığını kavrar.**

- a) Otomatik sistemlerin sensörler, işlemciler, aktüatörler vb. temel bileşenleri tanıtılır.
- b) Sensörlerin çevresel veri toplamadaki rolü anlatılır ve bu verilerin otomatik sistem tarafından nasıl kullanıldığına örnek verilir.
- c) Veri işleme ve karar mekanizmaları anlatılır, işlemcilerin rolüne dikkat çekilir.
- ç) İşlemcilerin aldığı kararları fiziksel eylemlere dönüştüren aktüatörlerin (örneğin motorlar, vanalar) işlevlerine örnekler verilir.

**1.6. Yarı otonom ve otonom sistemlerin çalışma mantığını kavrar**

- Yarı otonom sistemlerin insan müdahalesine ihtiyaç duyan ancak belirli görevleri otomatik olarak gerçekleştiren sistemler olduğu ifade edilir.
- Otonom sistemlerin insan müdahalesine gerek duymadan belirli görevleri tamamen bağımsız olarak gerçekleştiren sistemler olduğu ifade edilir.
- Bu sistemlerde sensörlerin veri toplama, işlemcilerin karar alma ve aktüatörlerin eylem gerçekleştirme işlevlerine sahip oldukları anlatılır.
- Yarı otonom ve otonom sistemlerin, çeşitli alanlarda kullanılan örnekleri gösterilir.

**1.7. Otomatik, yarı otomatik ve otonom sistemler arasındaki farkı bilir.**

- Otomatik sistemlerin insan müdahalesi olmadan ama programlanmış talimatlar doğrultusunda önceden tanımlanmış koşullar altında çalıştığı anlatılır.
- Yarı otomatik sistemlerin belirli görevleri kısmen otomatik olarak gerçekleştirebilen, ancak insan müdahalesine ihtiyaç duyan sistemler olduğu anlatılır.
- Otonom sistemlerin ise programlanmış talimatlar veya insan müdahalesi olmaksızın bağımsız bir şekilde çevresel koşulları algılayarak elde ettiği verileri işleyip kararlar alabilen sistemler olduğu anlatılır.
- Seçilen bir sorun için yarı otomatik, otomatik ve otonom bir sistem ile çözüm bulmak için beyin fırtınası yapılır.

**1.8. Öğrenebilen sistemlerin çalışma mantığını kavrar.**

- Öğrenebilen sistemlerin çevresel verileri toplayıp işleyerek gelecekteki kararlarını iyileştirebilen sistemler olduğu anlatılır.
- Makine öğrenmesi ve yapay zekâ terimleri anlatılarak öğrenebilen sistemleri ile bağlantısı kurulur.
- Veri madenciliği, derin öğrenme ve karar ağaçları gibi öğrenme algoritmalarının öğrenebilen sistemlerin temelini oluşturmasına dikkat çekilir.
- Öğrenebilen sistemlerin tahminlerde bulunup kararlar alabilmesi için genellikle büyük veriler üzerinde çalışması gerektiğine dikkat çekilir.

**Değerler**

Öz denetim (Kazanım 1.1., 1.2., 1.3., 1.7.)

Sorumluluk (Kazanım 1.3., 1.7.)

Yardımselverlik (Kazanım 1.3., 1.7.)

**Alan Becerileri**

Gözlem Yapma Becerisi (Kazanım 1.1., 1.2., 1.5., 1.6.)

Tahmin Etme Becerisi (Kazanım 1.1., 1.3., 1.7.)

Soyutlama Becerisi (Kazanım 1.3., 1.7.)

Algoritmik Düşünme Becerisi (Kazanım 1.3., 1.7.)

Sistem Tasarlama Becerisi (Kazanım 1.3., 1.7.)

## 2. ÜNİTE: ROBOTİK BİLEŞENLER

### Ünite Açıklaması

Bu ünite, robotik teknolojinin temel bileşenlerini, kodlama ve tasarım süreçlerini, farklı motor türlerini ve hareket nesnelerini öğrencilere tanıtarak robotik projeleri oluşturma ve sorunları çözme yeteneklerini geliştirmeye odaklanır.

### Kazanım ve Açıklamaları

#### 2.1. Yazılım ve sistem tasarımına uygun akış diyagramı tasarlar.

- Akış diyagramlarının robotik projelerin yazılım süreçlerini görselleştirmede kullanıldığı anlatılır.
- Akış diyagramlarının kodun mantığını anlamada ve tasarım hatalarını önceden tespit etmede ne kadar önemli olduğuna dikkat çekilir.
- Öğrenciden, bir problemin çözümü için akış diyagramı oluşturması istenerek karmaşıklığı yönetme becerisi kazanması sağlanır.
- Bir kodlama ekibiyle çalışırken akış diyagramlarını kullanmanın iş birliği içinde çalışmayı kolaylaştıracağı anlatılır.

#### 2.2. Kodların anlamını açıklar.

- Bir kodlama dili için syntax, değişkenler, veri türleri, ifadeler, operatörler, kontrol yapıları, fonksiyonlar, kütüphaneler, yorumlar ve hata ayıklama kavramları açıklanır.
- Örnek bir kod bloğunun ne anlama geldiği ve nasıl çalıştığı anlatılır.
- Kodların mantığını anlamının, hata ayıklama ve kodu geliştirme süreçlerine etkisine dikkat çekilir.

#### 2.3. Kodları amacına uygun düzenler.

- Yazılan kodun düzenli olmasının, programın performansına etkisine ve daha sonrada kodun bakımını kolaylaştırabileceğine dikkat çekilir.
- Temiz ve düzenli kod yazmanın, özellikle iş birliği ile yazılan projelerin daha iyi yönetilebilmesine yardımcı olduğuna değinilir.
- İçinde hatalı kodların bulunduğu örnek bir uygulama ile hataların tespiti, kodların düzeltilmesi ve sadeleştirilmesi gibi aşamalar farklı öğrencilere yaptırılır. Bu sayede öğrencilere, bir projenin ekip çalışmasıyla düzenli olarak nasıl yürütüldüğü gösterilmiş olur.

#### 2.4. Temel elektronik kavramları ve bileşenleri açıklar.

- Temel elektronik kavramları (direnç, transistör, devreler) ve bileşenler (LED'ler, sensörler) anlatılır.
- Elektronikğin temellerini iyi kavramanın, robotik projelerinde elektronik devre oluşturmayı kolaylaştıracağına dikkat çekilir.
- Elektronik devreyi tasarlamamanın robotik projelerin en önemli aşamalarından biri olduğu vurgulanır. Bu aşamada yapılan hataların projenin diğer aşamalarında sürekli sorun yaşanmasına sebep olacağına

dikkat çekilir.

**2.5. Elektronik eklentileri belirler.**

- a) Bir robotik proje için hangi elektronik bileşenlerin gerektiğini belirleme süreci anlatılır.
- b) Elektronik bileşenlerin projelerindeki rollerini anlatarak donanım seçimi hakkında fikir sahibi olmaları sağlanır.

**2.6. Problemleri robotik kodlama ile çözebilmek için mantık geliştirir.**

- a) Robotik projelerde karşılaştıkları problemleri tanımlama ve bu problemleri çözmek için mantık geliştirme becerisi kazandırılır.
- b) Örnekler üzerinden programlama ve kodlama yoluyla problemlere sistematik bir şekilde yaklaşmayı öğrenmeleri için pratik yapma fırsatı sunulur.

**2.7. Sensörler kullanarak yeni sistemler önerir.**

- a) Robotikte en çok kullanılan sensörlerin neler olduğu hakkında bilgi verilir.
- b) Farklı sensörlerin kullanıldığı robotik proje örnekleri gösterilir.
- c) Verilen bir problemin çözümü için kullanılacak alternatif sensörler gösterilerek farklı çözüm yollarına ulaşabileceklerinin bağlantısı kurulur.

**2.8. Modüller kullanarak yeni sistemler geliştirir.**

- a) Modüllerin ne olduğu ve nasıl kullanıldığı anlatılır.
- b) Modüllerin robotik projelerde hızlı prototipleme ve sistemlerin genişletilmesi için nasıl kullanılacağı örneklerle gösterilir.
- c) Verilen bir probleme yönelik öğrencinin uygun modülü tespit etmesi ve bu modülü kullanarak bir çözüm üretmesi istenir.

**2.9. Motor türlerini kullanarak özgün sistemler geliştirir.**

- a) Farklı türdeki motorların (DC motorlar, servo motorlar, step motorlar) ne zaman ve nasıl kullanılacağı anlatılır.
- b) Motorların hareket kontrolünün motor sürücüler ile nasıl yapıldığı ve robotik projelerdeki rolleri açıklanır.
- c) Birden fazla motor türünü bir arada kullanan bir robotik sistem geliştirmeleri istenir.

**2.10. Hareketli nesnelerin kullanıldığı farklı sistemler geliştirir.**

- a) Hareketli nesnelerin (dişliler, tekerlekler, pervaneler vb.) farklı robotik sistemlerde nasıl kullanıldığı örneklerle gösterilir.
- b) Hareketli nesnelerin farklı tasarım ve hareketlilik gereksinimlerini nasıl karşılayabileceği örneklerle gösterilir.
- c) Verilen bir problemin çözümüne yönelik, hareketli nesnelerin kullanıldığı robotik bir çözüm istenerek öğrencilerin hareketli nesneleri etkili bir şekilde kullanması sağlanır.

**2.11. Elektronik tasarımın kodlarını hazırlar.**



- Geliştirilen elektronik tasarımın kodlarını yazma süreci ve bu kodların donanımla nasıl etkileşimde bulunduğu açıklanır.
- Kodlama ve donanım bileşenlerini uyumlu bir şekilde kullanarak verimli robotik projeleri nasıl oluşturacakları öğretilir.
- Öğrencilere hazırlanmış bir elektronik devre verilir ve istenen uygulama anlatılarak bu elektronik devre için kodları yazması istenir.

### Değerler

Öz denetim (Kazanım 2.1., 2.6., 2.7., 2.9.)

Sorumluluk (Kazanım 2.3., 2.6., 2.9.)

### Alan Becerileri

Gözlem Yapma Becerisi (Kazanım 2.2., 2.7., 2.10.)

Tahmin Etme Becerisi (Kazanım 2.1., 2.2., 2.9.)

Soyutlama Becerisi (Kazanım 2.1., 2.9.)

Algoritmik Düşünme Becerisi (Kazanım 2.1., 2.6., 2.9., 2.11.)

Sistem Tasarlama Becerisi (Kazanım 2.7., 2.8. 2.9., 2.10., 2.11.)

## 3. ÜNİTE: PROJE GELİŞTİRME

### Ünite Açıklaması

Bu ünite öğrencilere, ileri düzeyde robotik projeler tasarlama ve geliştirme sürecini öğretmeye odaklanmaktadır. Öğrenciler, çeşitli ortam algılayıcı sensörler, motor türleri, iletişim ekipmanları ve ileri seviye sensörler gibi robotik bileşenleri kullanarak özgün projeler oluşturmayı öğrenirler. Ayrıca, hareketli parçaların ve farklı bileşenlerin nasıl entegre edileceği konuları da bu ünite içinde ele alınır. Bu sayede öğrenciler, karmaşık ve yenilikçi robotik projeler tasarlama ve geliştirme becerilerini kazanırlar.

### Kazanım ve Açıklamaları

#### 3.1. Ortam algılayıcı sensörler ile uygulamalar geliştirir.

- Öğrencilere, verilen bir problemin çözümü için hangi sensörü seçmesi gerektiği hakkında bilgi verilir.
- Ortam algılayıcı sensörleri kullanarak farklı uygulamalar geliştirmeyi öğrenirler.
- Doğru sensör kullanımı ile çevresel verileri toplama ve analiz etme yeteneklerini geliştirmelerine katkı sağlanır.

#### 3.2. Farklı motor türleri ve sürücüler ile projeler geliştirir.

- Farklı motor türlerinin (örneğin DC motorlar, servo motorlar) nasıl bir arada kullanılabileceği anlatılır.
- Motor sürücülerin robotik projelerdeki rolü açıklanır ve farklı motorları ayrı ayrı veya bir arada sürmek için hangi motor sürücülerin kullanılması gerektiği örneklerle gösterilir.

- c) Aynı motor türlerini bir arada kullanarak veya dişli kutuları kullanılarak daha yüksek güç elde edilebileceğine dikkat çekilir.
- ç) Uygun motor sürücü ile aynı veya farklı türde birden fazla motor bir arada kullanılarak öğrencilere örnek uygulama yaptırılır.

### 3.3. Hareketli parçalar ile robotik üniteler oluşturur.

- a) Hareketli parçaların (örneğin tekerlekler, paletler) robotik ünitelerin hareketliliği için nasıl etkili kullanılabileceğine dair örnekler gösterilir.
- b) Hareketli parçaların kullanıldığı farklı robotik sistemlerin tasarım ve montajını yapmayı öğrenirler.
- c) Hareketli parçaları entegre ederek özgün bir robotik proje geliştirmelerine yardımcı olunur.

### 3.4. İletişim ekipmanlarının kullanıldığı sistemler oluşturur.

- a) İletişim ekipmanlarının (örneğin Bluetooth, Wi-Fi modülleri) robotik projelerde nasıl kullanılabileceği açıklanır.
- b) İletişim ekipmanlarını kullanarak veri transferi yapması ve robotları uzaktan kontrol etmeyi sağlayacak sistemleri nasıl tasarlayacağı öğretilir.
- c) İletişim protokollerinin arasındaki fark karşılaştırmalı olarak gösterilir.
- ç) İki mikrodenetleyicinin veya bir kontrolcü ile bir mikrodenetleyicinin kendi aralarında haberleştiği bir sistem geliştirmeleri istenir.

### 3.5. İleri seviye sensörlerin kullanıldığı özgün projeler hazırlar.

- a) İleri seviye sensörlerin (örneğin, görüntü işleme kameraları, lazer sensörler) nasıl çalıştığı ve kullanıldığı öğretilir.
- b) Öğrencilere bir problem verilir. Çözümü için ileri seviye sensörlerin kullanıldığı özgün bir robotik proje geliştirmeleri istenir ve bu süreçte öğrencilere rehberlik edilir.
- c) Öğrencilerden kendi yaşam alanlarını düşünerek bir problem tespit etmeleri istenir. Çözümü için ileri seviye sensörlerin kullanıldığı özgün bir robotik proje geliştirmeleri istenir ve bu süreçte öğrencilere rehberlik edilir.

#### Değerler

Sabır (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5.)

Sorumluluk (Kazanım 3.5.)

Yardımselverlik (Kazanım 3.5.)

#### Alan Becerileri

Sistem Tasarlama Becerisi (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5.)

Sistem Kurma (İnşaa Etme) Becerisi (Kazanım 3.5.)

Test Etme ve Hata Ayıklama Becerisi (Kazanım 3.5.)

Teknik İşlem Becerisi (Kazanım 3.5.)

## MOBİL UYGULAMA GELİŞTİRME

Mobil uygulama geliştirme konusundaki temel beceri ve bilgilerin geliştirilmesi, mobil uygulama geliştirme sürecinin geçmişten bugüne öneminin öğretilmesi, mobil uygulama geliştirme tasarım ve kodlama süreçlerindeki aşamaların öğretilerek öğrenciler tarafından mobil uygulamalar geliştirilmesi amaçlanmaktadır.

### 1. ÜNİTE: MOBİL PROGRAMLAMA OKURYAZARLIĞI

#### Ünite Açıklaması

Bu ünite; sırasıyla mobil cihaz ve uygulamaların tarihsel gelişimi, mobil teknolojilerin ve mobil uygulamaların günümüzdeki önemi, mobil uygulamaların kullanım alanları ve ülkemizde en çok kullanılan mobil uygulamalar üzerinde durulur. Mobil donanım bileşenleri ile mobil cihazlarda yer alan sensörlerin çalışma mantığı anlatılır. Mobil işletim sistemlerine değinildikten sonra mobil işletim sistemlerinin benzer ve farklı yönleri tartışılır. Uygulama geliştirirken kullanılan farklı tasarım yapılarına değinilir.

#### Kazanım ve Açıklamaları

##### 1.1. Mobil cihaz ve uygulamaların tarihsel gelişimini kavrar.

- Mobil cihaz ve uygulamaların tarihsel gelişiminde yer alan önemli adımlar listelenir.
- Mobil cihaz ve uygulamaların tarihsel gelişiminde yer alan önemli adımlar örneklerle açıklanır.

##### 1.2. Mobil uygulamaların ve mobil teknolojilerin günümüzdeki önemini kavrar.

Mobil uygulamaların ve mobil teknolojilerin günümüzdeki önemi nedenleriyle anlatılır.

##### 1.3. Mobil Uygulamaların kullanım alanlarını sıralar.

Mobil uygulamaların; bankacılık, alışveriş, içerik paylaşımı, okuma-yazma, harita, ulaşım, sağlık ve eğitim gibi birçok alanda kullanımı örneklerle anlatılır.

##### 1.4. Ülkemizde en çok kullanılan ve topluma yararlı olan yerli mobil uygulamaları tanıır.

- Ülkemizde en çok kullanılan ve topluma yararlı yerli mobil uygulamalar üzerinde durulur.
- Yerli mobil uygulamalar ile toplumun özel gereksinimlerine çözümler bulunması üzerinde durulur.
- Ülkemize ait E-devlet hizmetleri, mobil bankacılık uygulamaları, sağlık takip uygulamaları, mesajlaşma uygulamaları ve e-ticaret platformları gibi uygulamalar ile ülkemizin dijital dönüşümünün desteklen-diği vurgulanır.
- Ülkemizde en çok kullanılan ve topluma yararlı yerli mobil uygulamaların tercih nedenleri açıklanır.
- Yerli mobil uygulamaların ülkemizin sürdürülebilir bir kalkınma sağlaması açısından önemli olduğu üzerinde durulur.

##### 1.5. Mobil donanım bileşenlerinin çalışma mantığını açıklar.

- Mobil donanım bileşenleri; temel donanım bileşenleri, çoklu ortam bileşenleri olmak üzere iki gruba ayrılarak bileşenler tanıtılır.
- Temel donanım bileşenlerinden anakart, RAM ve işlemcinin görevleri ve çalışma mantığı anlatılır.

**1.6. Mobil cihazlarda yer alan sensörlerin çalışma mantığını kavrar.**

- Jiroskop, manyetik ölçer, yakınlık, ışık, basınç, parmak izi ve hareket sensörlerine değinilir.
- Mobil cihazdaki sensörlerin sökölüp takılması gösterilir.

**1.7. Mobil işletim sistemlerini sıralar.**

- Mobil cihazlarda kullanılan işletim sistemleri örneklerle açıklanır.
- Mobil işletim sistemlerinden en çok kullanılan işletim sistemlerinin özellikleri üzerinde durulur.
- Mobil işletim sistemlerinden en çok kullanılan işletim sistemlerinin maliyet, güvenlik, hız ve kişiselleştirilebilirlik bakımından karşılaştırması yapılır.

**1.8. Mobil uygulama geliştirme tasarım yapılarını açıklar.**

- Mobil uygulama geliştirme tasarım yapıları; Yerel (Native) mobil platform, Duyarlı (Responsive) mobil platform, Melez (Hibrit) mobil platform ve Karma (Cross) mobil platform olarak dört gruba ayrılarak tanımlanır.
- Yerel (Native) mobil platform, Duyarlı (Responsive) mobil platform, Melez(Hibrit) mobil platform ve Karma (Cross) mobil platformların avantaj ve dezavantajlarından bahsedilir.
- Karma mobil platformun diğer platformlara göre üstün olma nedenleri üzerinde durulur.

**1.9. Mobil uygulama geliştirme araçlarını kavrar.**

Karma mobil platform uygulama araçlarından açık kaynaklı kütüphanelerin özellikleri üzerinde durulur.

**Değerler**

Sorumluluk (Kazanım 1.6.)

Vatanseverlik (Kazanım 1.4.)

Saygı (Kazanım 1.2., 1.3., 1.4., 1.7., 1.10.)

**Alan Becerileri**

Soyutlama Becerisi (Kazanım 1.1., 1.3., 1.4., 1.7., 1.8., 1.9., 1.10.)

Mantıksal Düşünme Becerisi (Kazanım 1.2., 1.4., 1.5., 1.6., 1.7., 1.8., 1.10.)

**2. ÜNİTE: MOBİL UYGULAMA TASARIMI VE ARAYÜZÜ****Ünite Açıklaması**

Bu ünite; sırasıyla hedef kitleyi belirlemenin önemi, mobil uygulamanın basit ve kullanılabilir olmasının gerekliliği, mobil uygulamaların farklı cihazlar ve farklı işletim sistemlerine uyumluluğunun önemi üzerinde durulur. Uygulama tasarımlarında dikkat edilmesi gereken görsel-estetik ilkeler anlatılır. Mobil uygulama geliştirme programının sistem gereksinimlerinden bahsedilir. Mobil uygulama geliştirme programının kurulumu, emülatör kurulumu gösterilir. Mobil uygulama geliştirme programının arayüz bileşenleri tanıtılır.

**Kazanım ve Açıklamaları**

**2.1. Mobil uygulama tasarımında hedef kitleyi belirlemenin önemini kavrar.**

- a) Uygulamanın başarısını arttırmak için hedef kitleyi tanımanın önemli olduğu vurgulanır.
- b) Hedef kitle özelliklerinin, beklentilerinin ve ihtiyaçlarının uygulama arayüz tasarımını belirlediği anlatılır.
- c) Hedef kitle belirleme adımlarından bahsedilir.

**2.2. Etkin bir mobil uygulamada olması gereken özellikleri açıklar.**

- a) Mobil uygulamanın basit, kolay ve kullanışlı olmasının gerekliliği anlatılır.
- b) Mobil uygulamanın basit bir tasarıma ve kullanışlı bir arayüze sahip olmasının, uygulamanın tercih edilmesinde ve kullanım oranını artırmasında önemli olduğu üzerinde durulur.

**2.3. Mobil uygulamanın farklı cihazlar ve farklı işletim sistemlerine uyumlu olmasının önemini açıklar.**

- a) Geniş bir kullanıcı kitlesine erişim sağlamak için farklı cihazlardaki ekran konfigürasyonlarına uyum sağlayabilecek esneklikte kullanıcı arayüzü geliştirmenin önemi anlatılır.
- b) Geniş bir kullanıcı kitlesine erişim sağlamak için uygulamanın tüm işletim sistemleriyle çalışacak şekilde geliştirilmesi gerekliliğinden bahsedilir.

**2.4. Mobil uygulama tasarım ilkelerini açıklar.**

- a) Mobil uygulama tasarımında dikkat edilmesi gereken estetik ilkeler nedenleriyle açıklanır.
- b) Mobil uygulama tasarımında şekil, renk, ton, boyut, yazı tipi, boşluk kullanımı gibi dikkat edilmesi gereken öğeler nedenleriyle anlatılır.

**2.5. Mobil uygulama geliştirme programının sistem gereksinimlerini açıklar.**

Mobil uygulama geliştirme programını bir bilgisayara kurabilmek için bilgisayarda olması gereken minimum işletim sistemi, disk alanı, işlemci, ekran çözünürlüğü ve RAM özellikleri anlatılır.

**2.6. Mobil uygulama geliştirme programının kurulumunu yapar.**

- a) Git yazılımının indirilmesi ve kurulum adımları uygulama yaptırılarak gösterilir.
- b) Flutter yazılımının indirilmesi ve kurulum adımları uygulama yaptırılarak gösterilir.
- c) Açık kaynaklı kodlama yazılımının indirilmesi ve kurulum adımları uygulama yaptırılarak gösterilir.

**2.7. Emülatör kurulumu yapar.**

Açık kaynaklı kodlama yazılımı üzerinden emülatör kurulum adımları uygulamalı gösterilir.

**2.8. Emülatör çalışma mantığını kavrar.**

Açık kaynaklı kodlama yazılımı üzerinden emülatörün çalıştırılması uygulamalı gösterilir.

**2.9. Mobil uygulama geliştirme programının arayüz bileşenlerini tanıır.**

Açık kaynaklı kodlama yazılımı üzerindeki menüler, klasör yapısı, kodlama alanı ve terminalden bahsedilir.

**Değerler**

Saygı (Kazanım 2.2., 2.3., 2.4.)

Sorumluluk (Kazanım 2.6., 2.7., 2.8.)

Sabır (Kazanım 2.6., 2.7., 2.8.)

Yardımselik (Kazanım 2.6., 2.7., 2.8.)

### **Alan Becerileri**

Soyutlama Becerisi (Kazanım 2.1., 2.5.)

Mantıksal Düşünme Becerisi (Kazanım 2.1., 2.2., 2.3., 2.4.)

Kodlama, Programlama Becerisi (Kazanım 2.6., 2.7., 2.8.)

## **3. ÜNİTE: PROGRAM BİLEŞENLERİ İLE ÇALIŞMA**

### **Ünite Açıklaması**

Bu ünite; temel bileşenlerin, yatay ve dikey yerleşim düzeni bileşenlerinin ve navigatör bileşeninin özellikleri üzerinde durulur. Sırasıyla temel bileşenlerin kullanımı ile ilgili basit mobil uygulama geliştirme, yatay ve dikey yerleşim düzeni bileşenlerinin kullanımı ile ilgili basit mobil uygulama geliştirme, temel bileşenler ile yatay ve dikey yerleşim düzeni bileşenlerinin birlikte kullanımı ile ilgili mobil uygulama geliştirme gösterilir. Temel bileşenlerle hazırlanmış bir sayfadan başka bir sayfaya geçiş yapma anlatılır. Bileşenlerin hepsini kullanmayı gerektiren karmaşık mobil uygulama geliştirme gösterilir.

### **Kazanım ve Açıklamaları**

#### **3.1. Mobil uygulamalarda kullanılan temel bileşenleri tanıır.**

Mobil uygulamada kullanılan Text, AppBar, Icon SizedBox, Image, Container gibi temel bileşenlerin en çok kullanılan özellikleri anlatılır.

#### **3.2. Temel bileşenleri kullanarak basit mobil uygulamalar geliştirir.**

- a) Text, AppBar, Icon SizedBox, Image, Container gibi temel bileşenlerin özelliklerini kullanarak basit mobil uygulama hazırlama gösterilir.
- b) Text, AppBar, Icon SizedBox, Image, Container gibi temel bileşenlerin özelliklerini kullanarak basit mobil uygulama hazırlatılır.

#### **3.3. Yatay ve dikey yerleşim düzeni bileşenlerini ve özelliklerini tanıır.**

Satır (row) ve sütun (column) yerleşim düzeni bileşenlerinin özellikleri anlatılır.

#### **3.4. Yerleşim düzeni bileşenlerini kullanarak basit mobil uygulamalar geliştirir.**

- a) Satır (row) ve sütun (column) yerleşim düzeni bileşenlerini kullanarak mobil uygulama hazırlama gösterilir.
- b) Satır (row) ve sütun (column) yerleşim düzeni bileşenlerini kullanarak mobil uygulama yaptırılır.

#### **3.5. Yerleşim düzeni bileşenleri arasındaki farkları ve kullanım yerlerini kavrar.**

- a) Yatay ve dikey yerleşim düzenleri arasındaki farklar anlatılır.
- b) Yatay ve dikey yerleşim düzenlerinin kullanıldığı yerler örneklerle gösterilir.

**3.6. Temel bileşenleri ve yerleşim düzeni bileşenlerini bir arada kullanır.**

- Text, image, container, icon temel bileşenleri ile row, column yerleşim düzeni bileşenlerinin bir arada kullanıldığı mobil uygulamaların hazırlanışı gösterilir.
- Text, image, container, icon temel bileşenleri ile row, column yerleşim düzeni bileşenlerinin bir arada kullanıldığı mobil uygulamalar yapılır.

**3.7. Temel bileşenler ile hazırlanmış birden çok sahneyle çalışır.**

- Butonlar, onay kutusu, textfield ve navigatör gibi temel bileşenler anlatılır.
- Temel bileşenlerle hazırlanmış bir sayfadan başka bir sayfaya geçiş yapma gösterilir.

**3.8. Çeşitli mobil uygulamalar geliştirir.**

- Temel bileşenler, yerleşim düzeni bileşenleri ile navigatör bileşenini birlikte kullanarak karmaşık mobil uygulama hazırlama gösterilir.
- Temel bileşenler, yerleşim düzeni bileşenleri ile navigatör bileşenini birlikte kullanarak karmaşık mobil uygulama hazırlatılır.

**Değerler**

Saygı (Kazanım 3.1., 3.3., 3.5.)

Sorumluluk (Kazanım 3.2., 3.4., 3.6., 3.7., 3.8.)

Sabır (Kazanım 3.2., 3.4., 3.6., 3.7., 3.8.)

Yardımselik (Kazanım 3.2., 3.4., 3.6., 3.7., 3.8.)

**Alan Becerileri**

Soyutlama Becerisi (Kazanım 3.1., 3.3., 3.5., 3.7.)

Mantıksal Düşünme Becerisi (Kazanım 3.5., 3.6., 3.7., 3.8.)

Algoritmik Düşünme Becerisi (Kazanım 3.2., 3.4., 3.6., 3.7., 3.8.)

Kodlama, Programlama Becerisi (Kazanım 3.2., 3.4., 3.6., 3.7., 3.8.)

Genelleme Becerisi (Kazanım 3.8.)

Test Etme ve Hata Ayıklama Becerisi (Kazanım 3.2., 3.4., 3.6., 3.7., 3.8.)

## PROGRAMLAMA DİLLERİ (JAVA)

Java programlama dilinde nesne yönelimli programlama teknikleri ile proje geliştirme becerilerinin kazandırılması amaçlanmaktadır.

### 1. ÜNİTE: JAVA PROGRAMLAMAYA GİRİŞ

#### Ünite Açıklaması

Bu ünite, Java programlama dilinin özellikleri üzerinde durularak öğrenciye neden bu programlama dilinin popüler olduğu kavratılır. Java programlama dilinin nesne tabanlı ve açık kaynak kodlu olması, geniş bir alanda ve yapay zekâ uygulamalarında kullanılabilirliğinin bulunması, dünya çapında yaygın ve geçerli olması özellikleri açıklanarak öğrenciye kavratılır. Gerekli kurulumların yapılması üzerinde durulur.

#### Kazanım ve Açıklamaları

##### 1.1. Java programlama dilini popüler kılan özellikleri kavrar.

- Nesne tabanlı ve açık kaynak kodlu bir programlama dili olduğu anlatılır.
- Mobil cihazlar, bilgisayarlar, ev aletleri gibi geniş bir alanda kullanılabildiği açıklanır.
- Yapay zekâ uygulamalarında kullanılabildiği anlatılır.
- Dünya çapında yaygın ve geçerli bir programlama dili olduğu vurgulanır.

##### 1.2. Java programlama için gerekli (Java Geliştirme Kiti ve Tümleşik Geliştirme Ortamı) kurulumları yapar.

- Java için JDK (Java Geliştirme Kiti) kurulumu gerçekleştirilir.
- IDE (Tümleşik Geliştirme Ortamı) kurulumu gerçekleştirilir.

#### Değerler

Sabır (Kazanım 1.2.)

Sorumluluk (Kazanım 1.2.)

#### Alan Becerileri

Bilimsel Süreç Becerisi (Kazanım 1.1.)

Bilgi İşlemsel Düşünme Becerisi (Kazanım 1.2.)

### 2. ÜNİTE: İLKEL VERİ TİPLERİ

#### Ünite Açıklaması

Bu ünite, Java programlamanın temelini oluşturan bilgiler verilecektir. Ekranda çıktı oluşturmak için metot kullanımı, dize değişmezi kullanımı, veri tiplerinin öğrenilerek amaca uygun olarak kullanımı yaptırılacaktır.

#### Kazanım ve Açıklamalar



## 2.1. Ekranda çıktı oluşturabilmek için gereken kodları kullanır.

“System.out.print” ve “System.out.println” kodlarını kullanarak ekrana yazı yazdırılır.

## 2.2. Dize değişmezleri oluşturur.

Dize değişmezi çift tırnak içine alınarak oluşturulur.

## 2.3. En uygun veri tipinin hangisi olduğunu belirler.

- Veri tipleri ilkel, referans veya null olarak sınıflandırılır.
- Sınıflandırılan veri tiplerinin hangi amaca uygun olduğu açıklanır.

## 2.4. Doğru türlerde değişkenleri kullanır.

- Integer, double, boolean veri tiplerinin kullanım amacı öğretilir.
- İlkel verileri temsil etmek için amaca uygun olan veri türü seçilerek değişkenler belirlenir.

### Değerler

Sorumluluk (Kazanım 2.1., 2.2., 2.3., 2.4.)

### Alan Becerileri

Bilimsel Süreç Becerisi (Kazanım 2.3., 2.4.)

Bilgi İşlemsel Düşünme Becerisi (Kazanım 2.1., 2.2., 2.4.)

## 3. ÜNİTE: SINIFLAR VE NESNELER

### Ünite Açıklaması

Bu ünite, nesne tabanlı programlamanın temel kavramları olan nesne ve sınıf tanımlanarak öğretilir. Public, private, yapılandırıcı metot kavramları öğretilir. Öğrencinin, sınıftan parametrelili ve parametresiz nesne oluşturması beklenir. Öğrenciye Void ve Non-Void Metot çağırımı yaptırılır.

### Kazanım ve Açıklamaları

#### 3.1. Nesne ve sınıf kavramlarını açıklar.

- Sınıfın nesne oluşturmak için kullanılan şablon olduğu açıklanır.
- Sınıfın yapısını etkileyen public, private kavramları fark ettirilir.
- Yapılandırıcı metot kavramı açıklanır.
- Nesnenin sınıfın bir örneği olduğu açıklanır.

#### 3.2. Sınıftan nesne oluşturur.

- Parametrelili yapılandırıcı ile nesne oluşturulur.
- Parametresiz yapılandırıcı ile nesne oluşturulur.

**3.3. Void ve Non-Void Metotları çağırır.**

- a) Void metodun ne olduğu tanıtılır.
- b) Void metot kullanımı yapılır.
- c) Non-Void Metodun ne olduğu tanıtılır.
- ç) Non-Void Metot kullanımı yapılır.

**Değerler**

Sorumluluk (Kazanım 3.2., 3.3.)

**Alan Becerileri**

Bilimsel Süreç Becerisi (Kazanım 3.1.)

Bilgi İşlemsel Düşünme Becerisi (Kazanım 3.2., 3.3.)

**4. ÜNİTE: KARŞILAŞTIRMA OPERATÖRLERİ, MANTIKSAL OPERATÖRLER VE KOŞULLU İFADELER****Ünite Açıklaması**

Bu ünite, karşılaştırma operatörleri tanıtılarak aritmetik ifadelerin değerlerini kıyaslama öğretilir. Mantıksal operatörler açıklanır ve örnek uygulama yaptırılır. Koşullara göre karar verme imkanı tanıyan koşullu ifadeler tanıtılarak uygulamalar yaptırılır.

**Kazanım ve Açıklamaları****4.1. Java programlama dili kurallarıyla karşılaştırma operatörleri ve mantıksal operatörler kullanır.**

- a) Aritmetik ifadelerin değerlerini kıyaslamak için karşılaştırma operatörleri (==, !=, <, >, <=, >=) ile örnek üzerinde açıklama ve uygulama yapılır.
- b) Mantıksal operatörler (&, |, ^, ||, &&, !) açıklanarak uygulama yapılır.

**4.2. Koşullu ifadeleri (if, If-else ve if else-if else) kullanır.**

- a) If koşullu ifadesi kullanımında mantıksal ifade doğru olduğu zaman yapı altında verilen kod parçasının çalıştığı anlatılır.
- b) If koşullu ifadesi ile uygulama yapılır.
- c) If-else koşullu ifadesi kullanımında mantıksal ifade doğru olduğu zaman if altındaki kod bloğunun yürütüleceğini, aksi takdirde ise else gövdesi altındaki kod bloğunun yürütüleceği anlatılır.
- ç) If-else koşullu ifadesi ile uygulama yapılır.
- d) Bir dizi seçenek olduğunda kullanılan çok yönlü seçimlerin (if-else if-else), her koşul için farklı ifadeler içeren bir dallanma yapısı olduğunda oluşturulacağı anlatılır.
- e) if else-if else koşullu ifadesi ile uygulama yapılır.

**Değerler**

Sorumluluk (Kazanım 4.1., 4.2.)

## Alan Becerileri

Bilimsel Süreç Becerisi (Kazanım 4.2.)

Bilgi İşlemsel Düşünme Becerisi (Kazanım 4.1., 4.2.)

## 5. ÜNİTE: DÖNGÜLER

### Ünite Açıklaması

Bu ünite, Java programlama dilinde kullanılan döngüler, iç içe döngü kavramı ve dallanma ifadeleri açıklanarak örnek uygulamalar yaptırılır.

### Kazanım ve Açıklamaları

#### 5.1. Farklı döngüleri (while, do-while, for) kullanarak program kodu yazar.

- a) while, do-while, for döngülerinin kullanım alanları açıklanır.
- b) Kullanım alanına göre uygun döngü seçilerek verilen problemi çözecek kodlar yazılır.

#### 5.2. İç içe döngüler kullanır.

- a) İç içe döngünün ne olduğu açıklanır.
- b) Verilen problemi iç içe döngü kullanarak çözecek kodlar yazılır.

#### 5.3. Dallanma ifadelerini (break, continue) kullanır.

- a) Döngü komutlarının akışını kontrol etmek için break ve continue ifadelerinin nerelerde kullanıldığı açıklanır.
- b) Örnekler içinde break ve continue ifadeleri kullanılır.

### Değerler

Sorumluluk (Kazanım 5.1., 5.2., 5.3.)

## Alan Becerileri

Bilimsel Süreç Becerisi (Kazanım 5.1., 5.3.)

Bilgi İşlemsel Düşünme Becerisi (Kazanım 5.1., 5.2., 5.3.)

## 6. ÜNİTE: ARRAY VE ARRAYLIST SINIFI

### Ünite Açıklaması

Bu ünite, sırasıyla; Array oluşturma, Array ile işlemler yapma, ArrayList ve ArrayList metodlarının kullanımı, ArrayList üzerinde dolaşım yapma ve algoritma oluşturma konuları hakkında açıklamalar ve uygulamalar yapılır.

## Kazanım ve Açıklamaları

### 6.1. Array (dizi) oluşturur.

- a) Dizi kullanımının, birden çok ilgili ögenin tek bir değişken kullanılarak temsil edilmesine olanak tanıdığı açıklanır.
- b) Dizinin boyutunun, oluşturma sırasında belirlendiği ve değiştirilmediği açıklanır.
- c) Dizilerin ilkel verileri veya nesne referans verilerini depolayabildiği açıklanır.
- ç) New anahtar kelimesi kullanarak dizi oluşturulur.

### 6.2. Array ile işlemler yapar.

- a) Dizi uzunluğunu bulunur.
- b) Dizi indis numaraları kullanarak erişim yapılır.
- c) Dizi elemanları sıralanır.
- ç) Aynı türde bulunan iki dizi karşılaştırılır.
- d) Dizi içerisinde arama yapılır.
- e) İhtiyaç hâlinde döngü kullanılır.

### 6.3. ArrayList kullanımını bilir.

- a) Array ve ArrayList farkı açıklanır.
- b) ArrayList tanımlama yöntemleri gösterilir.

### 6.4. ArrayList metotlarını kullanır.

- a) Add, get, set, remove, clear, count, contains, sort, size metodlarının ne işe yaradığı anlatılır.
- b) Add, get, set, remove, clear, count, contains, sort, size metodları kullanılır.

### 6.5. ArrayList üzerinde dolaşım yapar.

Döngüler ArrayList içindeki tüm elemanlara erişmek için kullanılır.

## Değerler

Sorumluluk (6.1., 6.2., 6.3., 6.4., 6.5.)

## Alan Becerileri

Bilimsel Süreç Becerisi (Kazanım 6.3.)

Bilgi İşlemsel Düşünme Becerisi (6.1., 6.2., 6.3., 6.4., 6.5.)

## 7. ÜNİTE: KALITIM

### Ünite Açıklaması

Bu ünite de kalıtım yöntemi kavratılır. Superclass ve subclass kavramları açıklanarak uygulama yaptırılır. Overriding metodu, super anahtar kelimesi ve çok biçimlilik yöntemi kullanılır.

### Kazanım ve Açıklamaları

#### 7.1. Kalıtım yöntemini kullanır.

- Superclass ve subclass kavramlarının ne olduğu açıklanır.
- Superclass ve subclass oluşturulur.
- Overriding metodu kullanılır.
- Super anahtar kelimesi kullanılır.

#### 7.2. Çok biçimlilik yöntemini kullanır.

Çok biçimlilik kavramının ne olduğu açıklanarak örnek uygulama yapılır.

### Değerler

Sorumluluk (Kazanım 7.1., 7.2.)

### Alan Becerileri

Bilimsel Süreç Becerisi (Kazanım 7.1.)

Bilgi İşlemsel Düşünme Becerisi (Kazanım 7.1., 7.2.)

## 8. ÜNİTE: RECURSIVE (ÖZYİNELEMELİ) FONKSİYON

### Ünite Açıklaması

Bu ünite de, recursive fonksiyon açıklanarak ve örneklenerek tanıtılır. Öğrenciye recursive fonksiyon ile ilgili uygulama yaptırılır.

### Kazanım ve Açıklamaları

#### 8.1. Recursive fonksiyonun ne olduğunu kavrar.

Recursive fonksiyonun kendisini çağıran bir metot olduğu örneklerle açıklanır.

#### 8.2. Recursive fonksiyon kullanımı yapar.

Verilen problemi recursive fonksiyon kullanarak çözen kodlar yazılır.

### Değerler

Sorumluluk (Kazanım 8.2.)

### Alan Becerileri

Bilimsel Süreç Becerisi (Kazanım 8.1.)

Bilgi İşlemsel Düşünme Becerisi (Kazanım 8.2.)

## PROGRAMLAMA DİLLERİ (C++)

C++ programlama dilinde temel çalışma ortamı, karar - döngü yapıları, fonksiyonlar, diziler ve koleksiyonlar, form oluşturma ile veri tabanı işlemleri gibi konuları içeren becerilerin kazandırılması ve ayrıntılı biçimde ele alınması amaçlanmaktadır.

### 1. ÜNİTE: C++ ÇALIŞMA ORTAMI

#### Kazanım ve Açıklamaları

##### 1.1. C++ programlama dilini tanır.

- a) C++ programlama dilinin doğuş süreci ve neden ihtiyaç duyulduğu söylenir.
- b) Genel C++ programlama dili kavramları ve söz dizimi anlatılır.

##### 1.2. Kod düzenleyici arayüzünde temel işlemleri gerçekleştirir.

- a) C++ programlama dili arayüzünde kod yazma sürecinde hata ayıklama, otomatik tamamlama ve belge yönetimi araçları incelenir.
- b) Yeni bir C++ arayüzü projesi açar, proje dosyalarını düzenler, projenin yapılandırması, dosyaların eklenmesi ve düzenlenmesi işlemleri uygulanır.
- c) Yazdığı kodları düzenli bir şekilde yönetmesi, değişiklikleri kaydetmesi ve ardından derleyerek çalışan bir uygulama yapması anlatılır.

##### 1.3. Temel veri türlerini ve değişken işlemlerini açıklar.

- a) Temel veri türleri (int, string, double vb.) ve farklı tipteki veriler (tam sayılar, metin, ondalık sayılar vb.) incelenir.
- b) Program arayüzünde değişkenler oluşturulur, değerlerle ilişkilendirilir ve bu değerler güncellenir.
- c) Matematiksel operatörler (+, -, \*, / vb.) ile mantıksal operatörler (==, !=, <, > vb.) kullanılarak temel aritmetik ve karşılaştırma işlemleri gerçekleştirilir.
- ç) Bir değişkenin değeri diğer bir değişkene atanarak veya aralarında matematiksel işlemler yapılarak veri aktarımı ve dönüşümü gerçekleştirilir.

#### Değerler

Sabır (Kazanım 1.1., 1.3.)

Öz denetim (Kazanım 1.2.)

Saygı (Kazanım 1.1., 1.2., 1.3.)

#### Alan Becerileri

Soyutlama Becerisi (Kazanım 1.1., 1.2.)

Algoritmik Düşünme Becerisi (Kazanım 1.3.)

## 2. ÜNİTE: KARAR VE DÖNGÜ YAPILARI

### Kazanım ve Açıklamaları

#### 2.1. Karar ifadelerini örneklerle açıklar.

- Programın belirli koşullara göre farklı davranışlar sergilemesini sağlayan karar ifadelerini farklı durumlar altında kontrol edip programlarında belirli şartlara göre yönlendirmesi anlatılır.
- Karar ifadeleriyle ilgili karşılaştırmanın mantıksal operatörlerle (eşitlik, büyüklük-küçüklük kontrolü vb.) tanımlanması ve karşılaştırılması örneklendirilir.
- Programlamanın temel yapı taşı olan karar ifadelerini, kullanıcıdan alınan girdilere veya belirli şartlara bağlı olarak kullanacağı örnek uygulamalar yaptırılır.

#### 2.2. Mantıksal operatörleri kullanarak koşullar oluşturur.

- Mantıksal operatörler (AND, OR, NOT) anlatılır.
- İki veya daha fazla koşulun aynı anda doğru olması gerektiği durumlarda mantıksal operatörleri (AND, OR, NOT) birleştirerek kullanması sağlanır.
- Mantıksal operatörler karmaşık koşulların analizi için farklı örneklerde çalışılır.

#### 2.3. Döngü yapılarını kullanarak tekrarlayan işlemler gerçekleştirir.

- Döngü yapıları ile belirli işlemlerin tekrarlanması öğretilir.
- Farklı türdeki döngü yapılarının (for, while, do-while) çalışma prensibi açıklanır.
- Veri koleksiyonları üzerinde dolaşma, belirli bir aşamaya kadar işlem yapma, işlemi devam ettirme ve tekrar eden görevleri otomatikleştirme işlemlerinde döngüleri kullanma durumları örneklendirilir.

### Değerler

Sabır (Kazanım 2.1.)

Saygı (Kazanım 2.1., 2.2., 2.3.)

Öz denetim (Kazanım 2.2., 2.3.)

Yardımsellik (Kazanım 2.3.)

### Alan Becerileri

Soyutlama Becerisi (Kazanım 2.1., 2.3.)

Algoritmik Düşünme Becerisi (Kazanım 2.2.)

Veri Toplama, İşleme, Dönüştürme Becerisi (Kazanım 2.2., 2.3.)

## 3. ÜNİTE: FONKSİYONLAR VE DİZİLER

### Kazanım ve Açıklamaları

#### 3.1. C++ programlama dilinde kullanılan fonksiyonları tanıtır.

- a) C++ programlama dilinde fonksiyonların nasıl tanımlandığı ve çağrıldığı anlatılır.
- b) Fonksiyonların parametreleri iletilme ve değer döndürme işlemleri gösterilir.

### 3.2. Nesne tabanlı programlamayı anlar ve C++ programlama arayüzü içerisinde nasıl uygulandığını öğrenir.

- a) Nesne tabanlı programlamanın temel kavramları (sınıflar, nesnelere, kalıtım, çok biçimlilik) açıklanır ve nesne tabanlı programlamanın avantajları anlatılır.
- b) Nesne tabanlı programlama prensiplerini uygulayarak gerçek dünya problemlerini nasıl modelleyebileceği ile ilgili örnek uygulamalar gösterilir.

### 3.3. Değer ve referans tipleri arasındaki farkı kavrar.

- a) Değer tipleri (int, float, bool) ve referans tipleri (sınıflar, diziler) arasındaki temel farklar açıklanır.
- b) Değer tiplerinin değer kopyalama, referans tiplerinin referans kopyalama üzerinde nasıl çalıştığı açıklanır.
- c) Değer ve referans tiplerinin farklı kullanım senaryoları ve avantajları anlatılır.

### 3.4. Tek boyutlu ve çok boyutlu dizileri kullanır.

- a) Tek boyutlu ve çok boyutlu dizileri basit biçimde nasıl oluşturup kullanacağı anlatılır.
- b) Dizi elemanlarına nasıl erişileceği ve dizi işlemlerinin nasıl gerçekleştirileceği açıklanır.
- c) Dizilerin farklı veri tipleri ile nasıl kullanılacağı ve işlevselliği anlatılır.

### Değerler

Sabır (Kazanım 3.2., 3.3., 3.4.)

Öz denetim (Kazanım 3.3.)

### Alan Becerileri

Soyutlama Becerisi (Kazanım 3.1., 3.2., 3.3.)

Mantıksal Düşünme Becerisi (Kazanım 3.4.)

## 4. ÜNİTE: FORMLAR

### Kazanım ve Açıklamaları

#### 4.1. Formları, form sınıflarını, kontrol sınıflarını ve konteyner sınıflarını tanır ve bu yapıları kullanır.

- a) İşletim sistemi uygulamalarında kullanılan temel arayüz bileşenleri ve form sınıflarının nasıl oluşturulup kullanılacağı anlatılır.
- b) Kontrol sınıfları (butonlar, metin kutuları, etiketler vb.) tanımlanır ve bu kontrollerin form içine nasıl yerleştirileceği gösterilir.
- c) Konteyner sınıflarını (panel, grup kutusu, sekme kontrolü vb.) kullanarak formdaki bileşenleri düzenleme ve gruplama anlatılır.

#### 4.2. İletişim kutularını kullanır.



- MessageBox, OpenFileDialog, SaveFileDialog iletişim kutularını kullanıcılarla iletişim kurmak için eklemesi anlatılır.
- Kullanıcıya bilgi vermek, doğrulama yapmak veya hataları bildirmek için iletişim kutularını kullanarak kullanıcıdan veri alma ve uyarı mesajları ekleme örnekleri incelenir.
- Farklı senaryolar için iletişim kutularının mesaj içeriklerinin, başlıklarının ve düğme metinlerinin kişiselleştirilmesi uygulamaları ile dinamik ve kullanıcı dostu arayüzler oluşturulması durumları fark ettirilir.

### 4.3. Doğrulama ve veri bağlama işlemlerini öğrenir.

- Kullanıcı tarafından sağlanan verilerin doğruluğunu kontrol etmek için kullanıcı girişi doğrulama işlemleri (validasyon) icra ettirilir.
- Veri bağlama ve verileri kullanıcı arayüzüne bağlama işlemleri öğretilir.
- Kullanıcı arayüzü ile etkileşimli uygulamalar geliştirmek için doğrulama ve veri bağlama örnekleri uygulanır.

### Değerler

Sabır (Kazanım 5.2.)

Öz denetim (Kazanım 5.3.)

### Alan Becerileri

Soyutlama Becerisi (Kazanım 5.1.)

Teknik İşlem Becerisi (Kazanım 5.2., 5.3.)

## 5. ÜNİTE: VERİ TABANI İŞLEMLERİ

### Kazanım ve Açıklamaları

#### 5.1. Veri tabanı yazılımlarını C++ arayüzü ile entegre eder.

- Veri tabanı yazılımlarını kurma ve C++ uygulamalarıyla entegre etme anlatılır.
- Projelerde uygun veri tabanını seçmek için veri tabanı yazılım türleri öğretilir.
- C++ programlama arayüzü ile veri tabanları arasında veri alışverişi yapma ve veri tabanı bağlantısını yönetme örnekleri incelenir.

#### 5.2. Veri tabanı tasarımı yapar.

- Veri tabanı tasarımı yapmayı ve veri tablolarını, alanlarını, ilişkilerini planlama öğretilir.
- Veri tabanı şeması oluştururken verilerin tutarlı ve erişilebilir olmasını sağlayacak durumlar vurgulanır.
- Farklı veri tabanı tasarım araçlarını kullanarak tasarım sürecini görselleştirme ve yönetme durumları örneklendirilir.

#### 5.3. Tablo işlemlerini gerçekleştirir.

- Veri tabanında tablo oluşturma, veri ekleme, güncelleme ve silme işlemlerini öğrenir.

- b) Veri tabanında oluşturulan tabloları sorgulama ve filtreleme yeteneklerini yönetme çalışılır.
- c) İşlevsel uygulamalar için tablo işlemlerini iş mantığıyla entegre etme durumu fark ettirilir.

**5.4. Temel SQL komutlarını kullanır.**

- a) Select, Insert, Update, Delete temel SQL komutlarını kullanarak verileri çekme, veri ekleme, var olan veriyi güncelleme ve silme işlemleri uygulanır.
- b) Veri tabanı sorguları oluştururken dinamik parametrelerle çalışma ve güvenli sorgu oluşturma becerileri geliştirilir.
- c) Karmaşık SQL sorguları (Join operatörleri, alt sorgular, indeksleme vb.) ile veri tabanlarını etkili bir şekilde sorgulama vurgulanır.

**5.5. İlişkisel veri tabanları oluşturur.**

- a) Farklı tablolar arasındaki ilişkileri çözmesi için ilişkisel veri tabanlarının kullanımı öğretilir.
- b) Veri tabanı modellemesi için veri tabanı şeması düzgün biçimde normalleştirilir.
- c) İlişkisel veri tabanlarının verileri tutma ve ilişkilendirme yeteneklerini kullanarak karmaşık veri tabanı yapıları oluşturulur.

**5.6. C++ programlama arayüzü ile oluşturulan veri tabanlarına bağlantı sağlar.**

- a) C++ programlama arayüzünde oluşturulan tablolardan veri çekme, veri güncelleme ve veri silme temel işlemleri uygulanır.
- b) Veri tabanı etkileşimli formlar tasarlatarak kullanıcıların arayüz üzerinden verilere erişme senaryoları gerçekleştirilir.

**Değerler**

Sabır (Kazanım 5.2., 5.4.)

Öz denetim (Kazanım 5.1., 5.3.)

Sorumluluk (Kazanım 5.1., 5.2., 5.3., 5.4., 5.5., 5.6.)

Saygı (Kazanım 5.1., 5.2., 5.3., 5.4., 5.5., 5.6.)

**Alan Becerileri**

Alan Becerileri

Soyutlama Becerisi (Kazanım 5.1., 5.3.)

Algoritmik Düşünme Becerisi (Kazanım 5.2., 5.4.)

Veri Toplama, İşleme, Dönüştürme Becerisi (Kazanım 5.2., 5.5.)

Veri Gösterimi ve Analizi (Kazanım 5.3., 5.4., 5.5.)

Model Doğrulama (Kazanım 5.6.)

Test ve Doğrulama (Kazanım 5.6.)

Sistem Entegrasyonu Becerisi (Kazanım 5.6.)

## PROGRAMLAMA DİLLERİ (C#)

C# programlama dili modülünde temel çalışma ortamı, karar - döngü yapıları, sınıflar, diziler ve koleksiyonlar ile form oluşturma gibi konuları içeren becerilerin kazandırılması hedeflenmektedir. Ayrıca, veri tabanı işlemlerinin ayrıntılı biçimde ele alınması amaçlanmaktadır.

### 1. ÜNİTE: C# ÇALIŞMA ORTAMI

#### Kazanım ve Açıklamaları

##### 1.1. C# programlama dilini tanır.

- C# programlama dilinin doğuş süreci ve neden ihtiyaca hâsıl olduğu söylenir.
- Genel C# programlama dili kavramları ve söz dizimi anlatılır.

##### 1.2. .NET Framework ile C# programlama dili ilişkisini tanımlar.

- .NET Framework ve temel bileşenleri (CLR, .NET sınıf kütüphanesi, CTS, CLS, managed code, Visual Studio IDE vb.) açıklanır.
- .NET Framework ile C# programlama dilinin .NET Framework tarafından sağlanan CLR üzerinde nasıl etkileşimde bulunduğu örneklendirilir.
- .NET Framework'ün C# programlama dili ile kullanılarak web tabanlı sitelerden büyük ölçekli iş uygulamalarına kadar geniş bir yelpazede uygulamalar geliştirildiği vurgulanır.

##### 1.3. Kod düzenleyici arayüzünde temel işlemleri gerçekleştirir.

- C# programlama dili arayüzünde kod yazma sürecinde hata ayıklama, otomatik tamamlama ve belge yönetimi araçları incelenir.
- Yeni bir C# programlama dili arayüzü projesi açma, proje dosyasını düzenleme, projeyi yapılandırma, dosya ekleme ve düzenleme işlemleri uygulanır.
- Yazdığı kodları düzenli bir şekilde yönetmesi, değişiklikleri kaydetmesi ve ardından derleyerek çalışan bir uygulama yapması anlatılır.

##### 1.4. Namespace kavramını bilir.

- Namespace kavramı ve kullanım alanları vurgulanır.
- Farklı namespace türlerini içe aktararak kullanma öğretilir.
- Namespace türleri kod organizasyonunda uygulanır.

##### 1.5. Temel veri türlerini ve değişken işlemlerini açıklar.

- Temel veri türleri (int, string, double vb.) ve farklı tipteki veriler (tam sayılar, metin, ondalık sayılar vb.) incelenir.
- Program arayüzünde değişkenler oluşturulur, değerlerle ilişkilendirilir ve bu değerler güncellenir.
- Matematiksel operatörler (+, -, \*, / vb.) ile mantıksal operatörler (==, !=, <, > vb.) kullanılarak temel aritmetik ve karşılaştırma işlemleri gerçekleştirilir.

- ç) Bir değişkenin değeri diğer bir değişkene atanarak veya aralarında matematiksel işlemler yapılarak veri aktarımı ve dönüşümü gerçekleştirilir.

## Değerler

Sabır (Kazanım 1.1., 1.2., 1.5.)

Öz denetim (Kazanım 1.3., 1.4.)

Saygı (Kazanım 1.1., 1.2., 1.3., 1.4., 1.5.)

## Alan Becerileri

Soyutlama Becerisi (Kazanım 1.1., 1.3.)

Algoritmik Düşünme Becerisi (Kazanım 1.2., 1.5.)

Veri Toplama, İşleme, Dönüştürme Becerisi (Kazanım 1.2., 1.4.)

## 2. ÜNİTE: KARAR VE DÖNGÜ YAPILARI

### Kazanım ve Açıklamaları

#### 2.1. Karar ifadelerini örneklerle açıklar.

- Programın belirli koşullara göre farklı davranışlar sergilemesini sağlayan karar ifadelerini farklı durumlar altında kontrol edip programlarında belirli şartlara göre yönlendirmesi anlatılır.
- Karar ifadeleriyle ilgili karşılaştırmanın mantıksal operatörlerle (eşitlik, büyüklük-küçüklük kontrolü vb.) tanımlanması ve karşılaştırılması örneklendirilir.
- Programlamanın temel yapı taşı olan karar ifadelerini, kullanıcıdan alınan girdilere veya belirli şartlara bağlı olarak kullanacağı örnek uygulamalar yaptırılır.

#### 2.2. Mantıksal operatörleri kullanarak koşullar oluşturur.

- Mantıksal operatörler (AND, OR, NOT) anlatılır.
- İki veya daha fazla koşulun aynı anda doğru olması gerektiği durumlarda mantıksal operatörleri (AND, OR, NOT) birleştirerek kullanması sağlanır.
- Mantıksal operatörler karmaşık koşulların analizi için farklı örneklerde çalışılır.

#### 2.3. Döngü yapılarını kullanarak tekrarlayan işlemler gerçekleştirir.

- Döngü yapıları ile belirli işlemlerin tekrarlanması öğretilir.
- Farklı türdeki döngü yapılarının (for, while, do-while) çalışma prensibi açıklanır.
- Veri koleksiyonları üzerinde dolaşma, belirli bir aşamaya kadar işlem yapma, işlemi devam ettirme ve tekrar eden görevleri otomatikleştirme işlemlerinde döngüleri kullanma durumları örneklendirilir.

## Değerler

Sabır (Kazanım 2.1.)

Saygı (Kazanım 2.1., 2.2., 2.3.)

Öz denetim (Kazanım 2.2., 2.3.)

Yardımselik (Kazanım 2.3.)

## Alan Becerileri

Soyutlama Becerisi (Kazanım 2.1., 2.3.)

Algoritmik Düşünme Becerisi (Kazanım 2.2.)

Veri Toplama, İşleme, Dönüştürme Becerisi (Kazanım 2.2., 2.3.)

## 3. ÜNİTE: SINIFLAR (CLASS)

### Kazanım ve Açıklamaları

#### 3.1. Nesne tabanlı programlamanın C# programlama dili içerisinde nasıl uygulandığını açıklar.

- C# programlama dili içinde sınıflar, nesnelere, kalıtım ve çok biçimlilik olarak ifade edilen nesne tabanlı yapıların gerçek dünya entegrasyonu sağlanarak kodun daha organize ve sürdürülebilir hâle gelmesini sağlayan avantajlar öğretilir.
- Nesne tabanlı programlama yapılarını kullanarak gerçek dünya sorunları incelenir.
- Nesne tabanlı programlama yapılarının prensipleri ve uygulamaların arka planındaki mantık açıklanarak örnek uygulamalar gerçekleştirilir.

#### 3.2. Kapsülleme, alanlar ve özellikler kavramlarını açıklar.

- Kapsülleme kavramı ve neden önemli olduğu anlatılır.
- Alanların özellikleri vurgulanır.
- Kapsüllemenin sınıf tasarımında kullanımı ve güvenli erişim yöntemleri açıklanır.

#### 3.3. Metotları kullanır.

- Belirli bir işlevi yerine getiren kod bloklarını tanımlamak için metotlar ve kod blokları anlatılır.
- Metotların parametre alması ve bu parametreleri kullanması vurgulanır.
- Metotların geri dönüş değerlerini ve sonuçları nasıl ilettiği incelenir.
- Instance metotlar, statik metotlar ve sanal metotlar arasındaki farklar açıklanır.

#### 3.4. Yapıcı ve yıkıcı metotların rolünü açıklar.

- Yapıcı ve yıkıcı metotların sınıfların süresini nasıl etkilediği incelenir.
- Yapıcı metotların parametreleri ve farklı aşırı yüklenmiş yapıcı metotların nasıl tanımlandığı anlatılır.
- Yıkıcı metotlarda kaynağın serbest bırakılması ve sınıfın sonlandırılması süreçlerindeki rolü açıklanır.

#### 3.5. Değer ve referans tipleri arasındaki farkı kavrar.

- Değer tipleri (int, float, bool) ve referans tipleri (sınıflar, diziler) arasındaki temel farklar anlatılır.

- b) Değer tiplerinin değer kopyalama, referans tiplerinin referans kopyalama üzerinde çalışması açıklanır.
- c) Değer-referans tiplerinin farklı kullanım senaryoları ve avantajları vurgulanır.

### 3.6. Kalıtım kavramını kavrar.

- a) Bir sınıfın diğer bir sınıftan kalıtım alması anlatılır.
- b) Alt sınıfların üst sınıflardan türetildiği ve temel sınıf ile türetilmiş sınıf arasındaki ilişki açıklanır.
- c) Kalıtım kavramının sınıf tasarımında kullanımı ve kodun yeniden kullanılabilirliği uygulamalarla incelenir.

### 3.7. Nesneye yönelik C# programlama dili temel yapılarını kullanır.

- a) Soyut sınıflar ve arayüzlerin kullanımı açıklanır.
- b) Çok biçimlilik ile farklı nesnelerin aynı metodları farklı şekillerde uygulayabilmesi incelenir.
- c) Statik sınıflar, isimsiz sınıflar, mühürlü sınıflar, parçalı sınıflar ve numaralandırmalar C# programlama arayüzünde çalıştırılır.

### Değerler

Sabır (Kazanım 3.1., 3.6., 3.7.)

Öz denetim (Kazanım 3.2., 3.3.)

Saygı (Kazanım 3.1., 3.3., 3.4., 3.5., 3.6., 3.7.)

### Alan Becerileri

Soyutlama Becerisi (Kazanım 3.1., 3.3., 3.4., 3.5., 3.6., 3.7.)

Mantıksal Düşünme Becerisi (Kazanım 3.2.)

Tahmin Etme Becerisi (Kazanım 3.3.)

Algoritmik Düşünme Becerisi (Kazanım 3.2., 3.3., 3.7.)

Kodlama, Programlama Becerisi (Kazanım 3.3.)

Genelleme Becerisi (Kazanım 3.3., 3.7.)

## 4. ÜNİTE: DİZİLER VE KOLEKSİYONLAR

### Kazanım ve Açıklamaları

#### 4.1. Tek boyutlu ve çok boyutlu dizileri kavrar.

- a) Tek boyutlu ve çok boyutlu dizilere değişkenleri doğru bir şekilde tanımlayarak değer atama uygulamaları C# programlama arayüzünde kullanılır.
- b) Verileri sıralama, arama ve değiştirme işlemleri için indeks numarası kullanarak dizi elemanlarına erişim uygulamaları gerçekleştirilir.
- c) Dizilerde sayıların, metinlerin ve diğer veri türlerinin kullanımı açıklanır.

#### 4.2. Koleksiyonlarda boxing ve unboxing kavramlarını açıklar.

- a) Değer tiplerinin (int, char, vb.) ve referans tiplerinin (object, interface, vb.) değişimi anlamına gelen boxing ve unboxing kavramları tanıtılır.
- b) Farklı tipteki öğeleri dinamik bir şekilde saklayabilen ArrayList koleksiyonunun oluşturulması ve dinamik verilerin depolanması anlatılır.
- c) Farklı tipteki verileri ArrayList koleksiyonu ile saklama açıklanır.
- ç) Koleksiyon elemanlarına erişme örnekleri çalışılır.
- d) Koleksiyonların verileri düzenlemek, filtrelemek ve hızlı erişim sağlamak için kullanılması anlatılır.
- e) Dinamik boyutları, veri yönetimi kolaylığı ve performans optimizasyonu özelliklerine sahip koleksiyonların avantajları vurgulanır.

#### **4.3. Koleksiyon türlerini kavrar.**

- a) List, queue-stack, dictionary, hashtable ve sortedlist koleksiyon türleri anlatılır.
- b) Her bir koleksiyon türünün kullanımı ve tercih edilme durumları incelenir.
- c) Veri saklama ve işleme gibi uygulama örnekleriyle koleksiyonların işlevselliği fark ettirilir.

#### **Değerler**

Öz denetim (Kazanım 4.2)

#### **Alan Becerileri**

Sorgulama Becerisi (Kazanım 4.2, 4.3)

Mantıksal Düşünme Becerisi (Kazanım 4.1)

## **5. ÜNİTE: FORMLAR**

### **Kazanım ve Açıklamaları**

#### **5.1. Formları, form sınıflarını, kontrol sınıflarını ve konteyner sınıflarını tanır.**

- a) Formları, işletim sistemi arayüzü uygulamalarında kullanılan düğmeler ve metin kutuları temel arayüz bileşenlerini kullanma anlatılır.
- b) Pencere oluşturma, diğer giriş kontrollerini form üzerine yerleştirme ve form sınıfları açıklanır.
- c) Kontrol sınıfları (butonlar, metin kutuları, etiketler vb.) ve bu kontrolleri bir form içine yerleştirme anlatılır.
- ç) Formdaki bileşenleri gruplamak amacıyla konteyner sınıflarının kullanılması, formdaki bileşenlerin düzenli bir şekilde yerleştirilmesi ve kullanıcı arayüzünün organize edilmesi çalışılır.

#### **5.2. Menüler, MenuStrip kontrolü ve ContextMenuStrip kontrollerini tanır.**

- a) Kullanıcı arayüzünü özelleştirmek için MenuStrip kontrolü ve ContextMenuStrip kontrolünün eklendiği örnekler uygulanır.
- b) Ana menüler ve alt menüler oluşturarak uygulamanın farklı bölümlerine hızlı erişim işlemi incelenir.

- c) Etkileşimi artırmak için kullanıcıların belirli nesnelere sağ tıkladığında açılan dinamik menülerin oluşturulması anlatılır.
- ç) Kontrol seçeneklerini özelleştirme ve kullanıcılar için etkileşimli menülerin entegre edilmesi C# programlama arayüzünde gerçekleştirilir.

### 5.3. İletişim kutularını kullanır.

- a) MessageBox, OpenFileDialog, SaveFileDialog iletişim kutularını kullanıcılarla iletişim kurmak için eklemesi anlatılır.
- b) Kullanıcıya bilgi vermek, doğrulama yapmak veya hataları bildirmek için iletişim kutularını kullanarak kullanıcıdan veri alma ve uyarı mesajları ekleme örnekleri incelenir.
- c) Farklı senaryolar için iletişim kutularının mesaj içeriklerinin, başlıklarının ve düğme metinlerinin kişiselleştirilmesi uygulamaları ile dinamik ve kullanıcı dostu arayüzler oluşturulması durumları fark ettirilir.

### 5.4. Doğrulama ve veri bağlama işlemlerini yapar.

- a) Kullanıcı tarafından sağlanan verilerin doğruluğunu kontrol etmek için kullanıcı girişi doğrulama işlemleri (validasyon) icra ettirilir.
- b) Veri bağlama ve verileri kullanıcı arayüzüne bağlama işlemleri anlatılır.
- c) Kullanıcı arayüzü ile etkileşimli uygulamalar geliştirmek için doğrulama ve veri bağlama örnekleri uygulanır.

### Değerler

Sabır (Kazanım 5.2., 5.4.)

Öz denetim (Kazanım 5.3.)

Sorumluluk (Kazanım 5.1., 5.2., 5.3., 5.4.)

Saygı (Kazanım 5.1., 5.2., 5.3., 5.4.)

### Alan Becerileri

Soyutlama Becerisi (Kazanım 5.1., 5.3.)

Algoritmik Düşünme Becerisi (Kazanım 5.2.)

Kullanıcı Arayüzü Tasarlama Becerisi (Kazanım 5.1., 5.2., 5.3., 5.4.)

Mantıksal Düşünme Becerisi (Kazanım 5.4.)

Veri Toplama, İşleme, Dönüştürme Becerisi (Kazanım 5.3., 5.4.)

## 6. ÜNİTE: VERİ TABANI İŞLEMLERİ

### Kazanım ve Açıklamaları

#### 6.1. Veri tabanı yazılımlarını C# arayüzü ile entegre eder.

- a) Veri tabanı yazılımlarını kurma ve C# uygulamalarıyla entegre etme anlatılır.



- b) Projelerde uygun veri tabanını seçmek için veri tabanı yazılım türleri açıklanır.
- c) C# programlama arayüzü ile veri tabanları arasında veri alışverişi yapma ve veri tabanı bağlantısını yönetme örnekleri incelenir.

**6.2. Veri tabanı tasarımı yapar.**

- a) Veri tabanı tasarımı yapmayı ve veri tablolarını, alanlarını, ilişkilerini planlama anlatılır.
- b) Veri tabanı şeması oluştururken verilerin tutarlı ve erişilebilir olmasını sağlayacak durumlar vurgulanır.
- c) Farklı veri tabanı tasarım araçlarını kullanarak tasarım sürecini görselleştirme ve yönetme durumları örneklendirilir.

**6.3. Tablo işlemlerini gerçekleştirir.**

- a) Veri tabanında tablo oluşturma, veri ekleme, güncelleme ve silme işlemleri anlatılır.
- b) Veri tabanında oluşturulan tabloları sorgulama ve filtreleme işlemlerine çalışılır.
- c) İşlevsel uygulamalar için tablo işlemlerini iş mantığıyla entegre etme durumu fark ettirilir.

**6.4. Temel SQL komutlarını kullanır.**

- a) Select, Insert, Update, Delete temel SQL komutlarını kullanarak verileri çekme, veri ekleme, var olan veriyi güncelleme ve silme işlemleri uygulanır.
- b) Veri tabanı sorguları oluştururken dinamik parametrelerle çalışma ve güvenli sorgu oluşturma becerileri geliştirilir.
- c) Karmaşık SQL sorguları (Join operatörleri, alt sorgular, indeksleme vb.) ile veri tabanlarını etkili bir şekilde sorgulama vurgulanır.

**6.5. İlişkisel veri tabanları oluşturur.**

- a) Farklı tablolar arasındaki ilişkileri çözmesi için ilişkisel veri tabanlarının kullanımı ifade edilir.
- b) Veri tabanı modellemesi için veri tabanı şeması düzgün biçimde normalleştirilir.
- c) İlişkisel veri tabanlarının verileri tutma ve ilişkilendirme yeteneklerini kullanarak karmaşık veri tabanı yapıları oluşturulur.

**6.6. C# programlama arayüzü ile oluşturulan veri tabanlarına bağlantı sağlar.**

- a) C# programlama arayüzünde oluşturulan tablolardan veri çekme, veri güncelleme ve veri silme temel işlemleri uygulanır.
- b) Veri tabanı etkileşimli formlar tasarlatarak kullanıcıların arayüz üzerinden verilere erişme senaryoları gerçekleştirilir.

**Değerler**

Sabır (Kazanım 6.2., 6.4.)

Öz denetim (Kazanım 6.1., 6.3.)

Sorumluluk (Kazanım 6.1., 6.2., 6.3., 6.4., 6.5., 6.6.)

Saygı (Kazanım 6.1., 6.2., 6.3., 6.4., 6.5., 6.6.)

**Alan Becerileri**

Soyutlama Becerisi (Kazanım 6.1., 6.3.)

Algoritmik Düşünme Becerisi (Kazanım 6.2., 6.4.)

Veri Toplama, İşleme, Dönüştürme Becerisi (Kazanım 6.2., 6.5.)

Veri Gösterimi ve Analizi (Kazanım 6.3., 6.4., 6.5.)

Model Doğrulama (Kazanım 6.6.)

Test ve Doğrulama (Kazanım 6.6.)

Sistem Entegrasyonu Becerisi (Kazanım 6.6.)

## PROGRAMLAMA DİLLERİ (DART)

Dart programlama dilinin temelleri, fonksiyonları, nesne tabanlı programlama, asenkron programlama, derleme süreci anlatılarak Dart programlama dilinde yazılım geliştirme becerilerinin kazanılması ile yazılım projeleri oluşturma yeteneklerinin geliştirilmesi amaçlanmaktadır.

### 1. ÜNİTE: DART PROGRAMLAMA DİLİ TEMELLERİ

#### Ünite Açıklaması

Bu ünite; dart programlama dilinin temel özellikleri, dart programlama dilindeki değişken tipleri, operatörler, fonksiyon ve parametreler, koşullu ifadeler ve döngüler üzerinde durulur. Dart programlama dili derleme süreci, derleme sürecinde AOT ve JIT arasındaki farklar, dart programlama dili paket yönetimi ile harici kütüphaneleri projeye ekleme anlatılır.

#### Kazanım ve Açıklamaları

##### 1.1. Dart programlama dilinin temel özelliklerini kavrar.

- Dart programlama dilinin geliştirilme nedeni ve kullanım alanları açıklanır.
- Dart programlama dilinin temel özellikleri, güçlü ve zayıf yönleri vurgulanır.

##### 1.2. Dart programlama dilindeki değişken tiplerini kavrar.

- Dart programlama dilinde kullanılan veri tipleri anlatılır.
- Değişkenlerin tanımlanması ve veri tiplerine atanması durumları açıklanır.

##### 1.3. Dart programlama dilindeki operatörleri öğrenir.

- Matematiksel, karşılaştırma ve mantıksal operatörlerin Dart programlama dilindeki kullanımı anlatılır.
- Operatörlerin öncelik sıralamaları ve gruplamaları örneklerle açıklanır.
- Dart programlama dilinde özel operatörler ve atama operatörlerinin kullanımı üzerinde durulur.

##### 1.4. Dart programlama dilinde fonksiyonları ve parametreleri tanımlar.

- Dart programlama dilinde fonksiyonların nasıl tanımlandığı ve çağırıldığı anlatılır.
- Fonksiyonların parametreleri iletme ve değer döndürme işlemleri gösterilir.
- Dart programlama dilinde anonim fonksiyonlar ve lambda ifadesinin önemi vurgulanır.

##### 1.5. Kontrol akışını yöneterek koşullu ifadeler ve döngüler kullanmayı kavrar.

- Dart programlama dilindeki koşullu ifadelerin (if-else, switch-case) kullanımı anlatılır.
- Dart programlama dilinde döngülerin (for, while, do-while) kullanımı anlatılır.
- Döngüler kullanılarak tekrarlı işlemler yapılabileceği vurgulanır.

##### 1.6. Dart programlama dilinde derleme sürecini kavrar.

- Dart programlama dilinde derleme ve çalışma süreci açıklanır.
- Dart programlama dilinde derleme sürecinin performans ve hız açısından oluşturduğu etkiler vurgulanır.

**1.7. Derleme aşamasında AOT ile JIT arasındaki farkları kavrar.**

- a) Dart programlama dilininin JIT ve AOT derleme arasında seçenekler sunmasının nedeni anlatılır.
- b) AOT derlemenin daha hızlı ve optimizasyonu yüksek kod üretme yeteneği olduğu vurgulanır.
- c) Hangi senaryolarda JIT ve AOT derleme seçilmesi gerektiği açıklanır.

**1.8. Dart programlama dilinde paket yönetimini kullanarak harici kütüphaneleri projelerine dâhil etmeyi kavrar.**

- a) Dart programlama dilinde paket yönetim sistemi (pub) ile projelere harici kütüphaneler eklenebileceği açıklanır.
- b) Harici kütüphanelerin projeye dahil edilmesi ve bağımlılıkların yönetilmesi anlatılır.

**Değerler**

Sorumluluk (Kazanım 1.6., 1.8.)

Saygı (Kazanım 1.1., 1.2., 1.3., 1.4., 1.5., 1.7., 1.10.)

Sabır (Kazanım 1.4., 1.6., 1.8.)

**Alan Becerileri**

Soyutlama Becerisi (Kazanım 1.1., 1.2., 1.3., 1.4., 1.5., 1.6., 1.7.)

Mantıksal Düşünme Becerisi (Kazanım 1.1., 1.3., 1.6., 1.7.)

Kodlama, Programlama Becerisi (Kazanım 1.4., 1.8.)

**2. ÜNİTE: NESNE TABANLI PROGRAMLAMA VE ASENKRON PROGRAMLAMA****Ünite Açıklaması**

Bu ünite; dart programlama dilinde nesne tabanlı programlama, sınıf ve nesne kavramları açıklanır. Constructor yöntemlerini, kalıtımı ve arayüzleri kullanarak program geliştirme ile dart programlama dilindeki istisna (exception) mekanizmasını kullanma anlatılır. Fonksiyonel programlama özelliklerini kullanarak yüksek düzeyde fonksiyonları yönetme ve dart programlama dilinde asenkron programlamanın önemi üzerinde durulur. Asenkron programlamada Future ve Stream kavramlarından bahsedilir. Asenkron programlamada async/await yapısını kullanma ile dart programlama dilinde dosya işlemleri yapma ve veri okuma/yazma işlemleri uygulamalı gösterilir.

**Kazanım ve Açıklamaları****2.1. Dart programlama dilinde nesne tabanlı programlama konusunu kavrar.**

- a) Nesne tabanlı programlamanın temel kavramları ve prensipleri açıklanır.
- b) Nesne tabanlı programlamanın önemi ve yazılım geliştirme sürecindeki avantajları vurgulanır.
- c) Dart programlama dilinde nesne tabanlı programlamanın nasıl kullanılacağı açıklanır.

**2.2. Sınıf ve nesne kavramlarını açıklar.**

- a) Dart programlama dilinde sınıf kavramı açıklanır.
- b) Nesnelerin sınıflardan nasıl türetildiği ve nesne tabanlı programlamanın nasıl çalıştığı açıklanır.
- c) Dart programlama dilinde sınıfların ve nesnelerin özellikleri ile metotlarının tanımlanma şekli açıklanır.

**2.3. Constructor yöntemlerini, kalıtımı ve arayüzleri kullanarak programlar geliştirir.**

- a) Sınıfların constructor metotlarını nasıl tanımlayacağı açıklanır.
- b) Kalıtım kavramı ve alt sınıfların üst sınıflardan nasıl türetildiği açıklanır.
- c) Dart programlama dilinde arayüz tanımlama ve uygulama işlemleri açıklanır.
- ç) Constructor ve kalıtım yöntemini kullanarak uygulama hazırlama gösterilir.

**2.4. Dart programlama dilindeki istisna (exception) mekanizmasını kullanmayı kavrar.**

- a) İstisna (Exception) kavramı açıklanır ve hata yönetiminin önemi vurgulanır.
- b) Dart programlama dilinde istisna yakalama ve işleme mekanizmaları açıklanır.

**2.5. Fonksiyonel programlama özelliklerini kullanarak yüksek düzeyde fonksiyonları yönetir.**

- a) Dart programlama dilinde fonksiyonların birinci sınıf nesnelere olduğu vurgulanır.
- b) Yüksek düzeyde fonksiyonlar, lambda ifadeleri ve işlevsel programlama özellikleri açıklanır.
- c) Fonksiyonel programlamanın faydaları ve kodları basitleştirmedeki önemi vurgulanır.

**2.6. Dart programlama dilinde asenkron programlamanın önemini kavrar.**

- a) Asenkron programlamanın açıklaması yapılır ve önemi vurgulanır.
- b) Dart programlama dilinde asenkron programlamanın uygulanma yöntemi açıklanır.
- c) İş parçacığı (thread) bloklamadan kaçınmanın ve kullanıcı arayüzü etkileşiminin geliştirilmesi açıklanır.

**2.7. Future ve Stream terimlerini kavrar.**

- a) Future ve Stream yapılarının asenkron programlamadaki kullanım şekli açıklanır.
- b) Asenkron işlemlerden gelecek sonuçları Future ve Stream ile nasıl işleneceği açıklanır.

**2.8. Asenkron operasyonların yönetimini sağlamak için async/await yapısını kullanmayı kavrar.**

- a) Async/await anahtar kelimelerinin anlamları açıklanır.
- b) Dart programlama dilinde asenkron async/await yapısını kullanılarak karmaşık kodların yönetilme süreci açıklanır.

**2.9. Dart programlama dilinde dosya işlemleri yapar.**

- a) Dosya işlemlerinin gerçekleştirilme şekli ve dosyalardan veri okuma-yazma işlemleri açıklanır.
- b) Dart programlama dilinde dosya işlemleri yapılırken hata yönetimi ve güvenlik konularının önemi vurgulanır.
- c) Dosya işlemleri kullanılarak verilerin depolanma ve okunma işlemlerinin örnekleri yaptırılır.

**Değerler**

Saygı (Kazanım 2.1., 2.2., 2.4 )

Sorumluluk (Kazanım 2.3., 2.5., 2.6., 2.7., 2.8., 2.9.)

Sabır (Kazanım 2.3., 2.5., 2.6., 2.7., 2.8., 2.9 )

Yardımselverlik (Kazanım 2.3., 2.5., 2.6., 2.7., 2.8., 2.9 )

## Alan Becerileri

Soyutlama Becerisi (Kazanım 2.1., 2.2., 2.4.)

Mantıksal Düşünme Becerisi (Kazanım 2.1., 2.2., 2.4., 2.5., 2.8.)

Kodlama, Programlama Becerisi (Kazanım 2.3., 2.5., 2.6., 2.7., 2.8., 2.9.)

Algoritmik Düşünme Becerisi (Kazanım 2.3., 2.5., 2.6., 2.7., 2.8., 2.9.)

Test Etme ve Hata Ayıklama Becerisi (Kazanım 2.3., 2.5., 2.6., 2.7., 2.8., 2.9.)

## 3. ÜNİTE: DERİNLEMESİNE DART PROGRAMLAMA DİLİ

### Ünite Açıklaması

Bu ünite; dart programlama dilinde reflection, metaprogramming kavramları, özel veri tipleri ve operatörler tanımlama, dart programlama dilinde yüksek performanslı uygulama hazırlama teknikleri anlatılır. Dart programlama dilindeki kod analiz araçlarını kullanma ve bu araçların geliştirme ortamına entegrasyonu ile dart programlama dilinde testler yazma gösterilir.

### Kazanım ve Açıklamaları

#### 3.1. Dart programlama dilinde reflection ve metaprogramming kavramlarını kavrar.

- Reflection ve metaprogramming kavramları açıklanır.
- Dart programlama dilinde bu kavramlar kullanılarak sınıfların ve nesnelerin manipüle edilebileceği açıklanır.

#### 3.2. Dart programlama dilinin genişletilebilirlik özelliklerini kullanarak özel veri tipleri ve operatörler tanımlar.

- Dart programlama dilinde özel veri tipleri oluşturma süreci açıklanır.
- Dart programlama dilinde operatör aşırı yükleme konsepti açıklanır ve özel operatör tanımlamanın önemi vurgulanır.
- Genişletilebilirlik özellikleri kullanılarak mevcut sınıfların genişletilerek yeni özellikler eklenilebileceği açıklanır.

#### 3.3. Dart programlama dilindeki kodları performanslı hâle getirmek için en uygun şekilde sokma tekniklerini uygular.

- Dart programlama dilinde performanslı kod yazmanın önemi vurgulanır ve en yaygın performans optimizasyon teknikleri açıklanır.
- Kodların çalışma sürelerini optimize etmek için lazy-loading gibi tekniklerin önemi vurgulanır.

c) Dart programlama dilindeki kod analiz araçları kullanılarak performans sorunlarının tespit edilebileceği vurgulanır.

### 3.4. Dart programlama dilinde mevcut olan kod analiz araçlarını ve IDE entegrasyonunu kullanmayı kavrar.

a) Dart programlama dilindeki kod analiz araçlarının (Dart Analyzer, Dart DevTools, vb.) kullanım şekli açıklanır.

b) Dart programlama dilinin geliştirme ortamlarında bu analiz araçlarının entegre edilme yöntemi açıklanır.

c) Kod analizi ve hata ayıklama süreçlerini daha verimli hâle getirmek için bu araçların gerekliliği vurgulanır.

### 3.5. Dart programlama dilinde test odaklı geliştirme (TDD) kavramını açıklar.

a) Test odaklı geliştirme kavramı açıklanır.

b) TDD yaklaşımı kullanılarak kod geliştirme sürecinin iyileştirilebileceği vurgulanır.

### 3.6. Dart programlama dilinde test yazmayı kavrar.

Dart programlama dilinde unit testler ve entegrasyon testler yazma kavramları açıklanır.

### Değerler

Saygı (Kazanım 3.1., 3.2., 3.3.)

Sorumluluk (Kazanım 3.4., 3.5.)

Sabır (Kazanım 3.4., 3.5.)

Yardımselik (Kazanım 3.4., 3.5.)

### Alan Becerileri

Soyutlama Becerisi(Kazanım 3.1., 3.2., 3.3., 3.5.)

Mantıksal Düşünme Becerisi(Kazanım 3.1., 3.3., 3.4.)

Algoritmik Düşünme Becerisi(Kazanım 3.4., 3.5.)

Kodlama, Programlama Becerisi (Kazanım 3.4., 3.5.)

Test Etme ve Hata Ayıklama Becerisi(Kazanım 3.4., 3.5.)

## PROGRAMLAMA DİLLERİ (PHP)

PHP programlama dilinde temel kavramlar, çalışma mantığı, yazım kuralları, fonksiyonlar ve veri tabanı işlemleri gibi konuların ayrıntılı biçimde ele alınması amaçlanmaktadır.

### 1. ÜNİTE: PHP TEMEL KAVRAMLAR

#### Ünite Açıklaması

Bu ünite, PHP programlama dilinin hangi ortamda çalıştığı ve nasıl kurulacağı konularına değinilmiştir.

#### Kazanım ve Açıklamaları

##### 1.1. Sunucu ve istemci tarafında çalışma mantığını kavrar.

- Sunucu ve istemci kavramları açıklanır.
- Sunucu tarafında çalışan yapılardan bahsedilir. Javascript kısaca açıklanır ve PHP programlama dili ile arasındaki farklar anlatılır.
- Sunucu ve istemci tarafında çalışan yapıların farkları anlatılır.

##### 1.2. PHP programlama dili için gerekli programları tanıır.

- PHP programlama dilinin çalışması için gerekli olan programlar tanıtılır.
- Gerekli olan programların işlevleri açıklanır.
- Her bir programın ne işlem yaptığından bahsedilir.

##### 1.3. PHP programlama dili için gerekli programların kurulumunu yapar.

- Bilgisayarda uygulamalı olarak gerekli programların kurulumu yaptırılır.
- Kurulum adımları açıklanarak, farklı yol ve yöntemlerin kullanılabileceğinden bahsedilir.

##### 1.4. PHP programlama dilinin farklı işletim sistemlerinde nasıl çalıştığını kavrar.

- PHP programlama dilinin çalışabildiği her bir işletim sisteminde nasıl çalıştığından bahsedilir.
- İşletim sistemlerinde PHP programlama dilinin çalışmasındaki farklar, avantajlar ve dezavantajlar anlatılır.

#### Değerler

Sabır (Kazanım 1.1., 1.3.)

Öz Denetim (Kazanım 1.3.)

#### Alan Becerileri

Soyutlama Becerisi (Kazanım 1.1.)

Teknik İşlem Becerisi (Kazanım 1.3.)



## 2. ÜNİTE: PHP PROGRAMLAMA DİLİNİ TANIYALIM

### Ünite Açıklaması

Bu ünite, PHP programlama dilinin çalışma mantığı, yazım kuralları ve yapılandırma ayarları konularına değinilmiştir.

### Kazanım ve Açıklamaları

#### 2.1. PHP programlama dilinin nasıl çalıştığını kavrar.

- PHP programlama dilinin nasıl çalıştığı görsel olarak anlatılır.
- Kullanıcının yapmak istediği iş ve işlemler hangi yollardan geçerek sonuca ulaştığı anlatılır.

#### 2.2. Yazım kurallarını kavrar.

- \$ karakterinin neden kullanıldığı anlatılır.
- Kod yazımında parantezlerin ve noktalama işaretlerinin nerelerde ve nasıl kullanılacağı açıklanır.
- Değişken tanımlama kuralları açıklanır.
- Veri türleri açıklanır.

#### 2.3. Yapılandırma ayarlarını uygular.

- Php.ini dosyasındaki ayarlar açıklanır.
- Öğrencilerden uygulamalı olarak php.ini dosyasında farklı ayarlar yapmaları istenir.

### Değerler

Sabır (Kazanım 2.1., 2.2.)

Sorumluluk (Kazanım 2.3.)

### Alan Becerileri

Gözlem Yapma Becerisi (Kazanım 2.1.)

Teknik İşlem Becerisi (Kazanım 2.3.)

## 3. ÜNİTE: PHP FONKSİYONLARI, ARİTMETİKSEL VE MANTIKSAL İŞLEMLER

### Ünite Açıklaması

Bu ünite, PHP programlama dilinde kullanılan matematiksel ve metinsel fonksiyonların kullanımı, koşul yapılarıyla mantıksal ve aritmetiksel işlemler, döngü yapılarının çalışma şekilleri konularına değinilmiştir.

### Kazanım ve Açıklamaları

#### 3.1. PHP programlama dili fonksiyonlarını tanır.

- Fonksiyonların işlevi anlatılır.
- Fonksiyonların kullanım gereksinimi nedenleriyle açıklanır.

- c) Fonksiyon kullanarak ve fonksiyon kullanmadan kod yazım örnekleri verilerek fonksiyonların avantajlarına değinilir.
- ç) Matematiksel ve metinsel fonksiyonlar anlatılır.

### 3.2. PHP programlama dilinde mantıksal ve aritmetiksel işlemleri kavrar.

- a) Karar yapıları anlatılır.
- b) Mantıksal ve aritmetiksel operatörler açıklanır.
- c) Örnekler verilerek hangi operatörün kullanılacağı sorulur.

### 3.3. Döngüleri kullanır.

- a) PHP programlama dilindeki döngü yapılarından bahsedilir.
- b) Aralarındaki farklar açıklanır.
- c) Örnekler verilerek hangi döngü yapısını kullanmalarının daha uygun olacağı sorulur.

### Değerler

Öz denetim (Kazanım 3.1., 3.2.)

Sorumluluk (Kazanım 3.1., 3.2., 3.3.)

### Alan Becerileri

Genelleme Becerisi (Kazanım 3.1., 3.2.)

Kodlama, Programlama Becerisi (Kazanım 3.2., 3.3.)

## 4. ÜNİTE: PHP PROGRAMLAMA DİLİ İLE VERİ TABANI İŞLEMLERİ

### Kazanım ve Açıklamaları

#### 4.1. Veri tabanı bağlantısı yapar.

MySQL dilinden bahsedilir. PHP programlama dili ile MySQL dilinin birlikte nasıl çalıştığı anlatılır. PHP programlama dilinde kullanılan MySQL dili komutları açıklanır. Veri tabanı bağlantısı örneklerle açıklanır. Hata ayıklama yolları gösterilir.

#### 4.2. Veri tabanı oluşturur.

PHP programlama dili komutları kullanarak veri tabanı, tablolar ve alanlar oluşturulur. Öğrencilerin veri tabanı, tablo ve alanlar oluşturmaları için gerekli kod satırlarını yazmaları istenir. Daha sonra yazdıkları kod satırları uygulamalı olarak çalıştırılarak sonuçları değerlendirilir.

#### 4.3. Veri tabanı üzerinde işlemler yapar.

INSERT, UPDATE, DELETE ve SELECT komutlarının kullanımları anlatılır. Hazır bir veri tabanı üzerinde nasıl çalıştıkları sonuçlarıyla değerlendirilir.

### Değerler

Sabır (Kazanım 4.1., 4.2., 4.3.)

Öz Denetim (Kazanım 4.1., 4.2., 4.3.)

**Alan Becerileri**

Algoritmik Düşünme Becerisi (Kazanım 4.2.)

Kodlama, Programlama Becerisi (Kazanım 4.1., 4.2.)

Kodlama, Programlama Becerisi (Kazanım 4.1.)

## PROGRAMLAMA DİLLERİ (JAVASCRIPT)

Javascript programlama dili temellerini, fonksiyonları, nesne tabanlı programlamayı, asenkron programlamayı, DOM ve Html'i birlikte kullanmayı, modüler kodlama işlemlerini öğrenerek, web geliştirme ve programlama becerilerinin geliştirilmesi amaçlanmaktadır.

### 1. ÜNİTE: JAVASCRIPT PROGRAMLAMA DİLİ TEMELLERİ

#### Ünite Açıklaması

Bu ünite; web geliştirme süreçlerinin işleyişi, Javascript programlama dilinin web sitelerindeki rolü, Javascript programlama dilinin terim ve kavramları üzerinde durulur. Javascript programlama dilinde değişkenler, veri tipleri, koşullu ifadeler, karşılaştırma operatörleri ve mantıksal operatörler kavramları anlatılır.

#### Kazanım ve Açıklamaları

##### 1.1. Web geliştirme süreçlerinin işleyiş şekillerini anlar.

- Web geliştirme süreçlerinin temel adımları olan, proje planlaması, tasarım, kodlama, test ve dağıtım süreçleri açıklanır.
- Süreçler arasında bağlantılar kurularak tasarımın kodlamaya dönüşme aşaması vurgulanır.

##### 1.2. Javascript programlama dilinin web sitelerindeki rolünü kavrar.

- Javascript programlama dilinin web sitelerindeki kullanım amacı açıklanır. Kullanıcı etkileşimini artırmak ve sayfaları dinamik hâle getirmek için kullanıldığı vurgulanır.
- Javascript programlama dilinin HTML ve CSS ile uyumlu bir şekilde çalıştığı açıklanır.

##### 1.3. Javascript programlama dilinin terim ve kavramlarını tanır.

- Javascript programlama dilinde kullanılan temel terimler ve kavramlar açıklanır.
- Değişken, fonksiyon, dizi gibi kavramların Javascript programlama dilindeki önemi vurgulanır.

##### 1.4. Değişkenleri Javascript programlama dili ile kullanır.

- Değişken tanımlama ve değişkenlerin kullanım şekli açıklanır.
- Değişkenlerin türleri açıklanır ve kullanılır.
- Değişkenlerin yaşam döngüsü anlatılır. Global ve lokal değişkenler arasındaki farklar açıklanır.

##### 1.5. Javascript programlama dili içerisinde kullanılan farklı veri tiplerini kavrar.

- Javascript programlama dilindeki veri tipleri (string, number, boolean, object, vb.) açıklanır.
- Veri dönüşümü ve tip dönüşümünün nasıl yapılacağı açıklanır.
- Javascript programlama dilindeki null ve undefined gibi özel değerler vurgulanır.

##### 1.6. Problemlerde karşılaşılan farklı senaryolarda koşullu ifadeleri kullanır.

- Javascript programlama dili ile if-else ifadeleri ve switch-case yapılarının kullanımı açıklanır.
- Koşullu ifadeler kullanılarak kullanıcı girişi doğrulama gibi senaryo örnekleri yapılır.

**1.7. Karşılaştırma operatörlerini Javascript programlama dili ile kullanır.**

- Javascript programlama dilinde karşılaştırma operatörleri (==, ===, !=, <, >, vb.) kullanılarak değişkenlerin karşılaştırması açıklanır.
- Karşılaştırma operatörleri ile koşullu ifadelerin birlikte kullanılmasıyla ilgili örnekler yapılır.

**1.8. Birden fazla koşul durumlarını birleştirmek için mantıksal operatörler kullanır.**

- Javascript programlama dilindeki mantıksal operatörler (&&, ||, !, vb.) açıklanır.
- Mantıksal operatörleri kullanarak birden fazla koşul birleştirilir ve karmaşık ifadeler oluşturulur.

**Değerler**

Sabır (Kazanım 1.1., 1.2., 1.3., 1.6.)

Özdenetim (Kazanım 1.4., 1.5., 1.6.)

Sorumluluk (Kazanım 1.4., 1.6., 1.7., 1.8.)

Yardımsellik (Kazanım 1.6.)

**Alan Becerileri**

Soyutlama Becerisi (Kazanım 1.1., 1.2., 1.3.)

Mantıksal Düşünme Becerisi (Kazanım 1.4., 1.5., 1.6., 1.7.)

Algoritmik Düşünme Becerisi (Kazanım 1.4., 1.5., 1.6., 1.7., 1.8)

Kodlama, Programlama Becerisi (Kazanım 1.1., 1.2., 1.3., 1.4., 1.5., 1.6., 1.7., 1.8.)

**2. ÜNİTE: JAVASCRIPT PROGRAMLAMA DİLİ FONKSİYONLARI VE NESNE TABANLI PROGRAMLAMA****Ünite Açıklaması**

Bu ünite; Javascript programlama dilinde fonksiyonlarının genel tanımı, fonksiyonların düzenlenmesi ve yönetilmesi, fonksiyonlarda kullanılan parametreler, fonksiyonların değişkenlere atanabilmesi üzerinde durulur. Javascript programlama dilinde diziler ve döngüler anlatılarak dizi elemanlarını sıralama ve filtreleme olaylarından bahsedilir. Nesne ve sınıf kavramları anlatılarak constructor'lara özellikler ekleme, nesnelere arası etkileşim kurma işlemleri uygulamalı olarak yapılır.

**Kazanım ve Açıklamaları****2.1. Javascript programlama dili ile fonksiyon tanımlar.**

- Javascript programlama dili ile fonksiyonların nasıl tanımlandığı ve kullanıldığı açıklanır.
- Fonksiyonlar kullanılarak tekrarlayan işlemlerin kolaylaştırıldığı vurgulanır.

**2.2. Fonksiyonları çağırarak kodun düzenlenmesini kavrar.**

- Fonksiyonların çağırılarak program içerisindeki kullanım şekli açıklanır.
- Fonksiyonların kodun daha düzenli ve okunabilir olması konusunda sağladığı katkı vurgulanır.

**2.3. Fonksiyonları çağırarak kodun yönetilmesini kavrar.**

- a) Fonksiyonların return değerini yakalaması ve bu değerın kullanılması açıklanır.
- b) Fonksiyonların farklı dosyalar içerisinde tutularak kod karmaşıklığının azaltılabileceğı vurgulanır.

**2.4. Fonksiyonlara parametre eklemeyi kavrar.**

- a) Fonksiyonlara parametre ekleme yöntemi ve bu parametrelerin kullanılma şekilleri açıklanır.
- b) Parametrelerin fonksiyonların kullanımında esnekliğı artırdığı vurgulanır.
- c) Fonksiyonların farklı tipteki parametreleri (sayı, dizi, nesne, vb.) işleyebilmesi açıklanır.

**2.5. Fonksiyonların değışkenlere atanarak kullanılabilceğini anlar.**

- a) Fonksiyonların birer değeri olduğunu ve bunun değışkenlere atanarak başka yerden çağrılabilceğı açıklanır.
- b) Fonksiyonların anonim (isimsiz) olarak kullanılabilceğı vurgulanır.
- c) Fonksiyonların return değerinin değışkenlere atanabilme durumu açıklanır.

**2.6. Javascript programlama dili ile dizileri tanımlar.**

- a) Javascript programlama dilinde dizilerin nasıl tanımlandığı ve elemanlarına nasıl erişileceğı açıklanır.
- b) Yeni dizi oluşturma, dizilere eleman ekleme ve çıkarma işlemlerinin nasıl yapılacağı açıklanır.

**2.7. Javascript programlama dilinde döngülerin kullanımını kavrar.**

- a) Javascript programlama dilindeki döngü yapıları (for, while, do-while) açıklanır.
- b) Döngülerin kullanıldığı senaryo örnekleri verilir.

**2.8. Diziler içerisinde gezinmeyi kavrar.**

- a) Diziler içerisinde döngüler kullanılarak nasıl gezileceğı açıklanır.
- b) Dizilerin uzunluğunun kontrol edilme durumu açıklanır.

**2.9. Diziler içerisinde işlem yapmayı kavrar.**

- a) Diziler içerisinde döngüler kullanılarak dizi elemanları üzerinde nasıl değışiklik yapılacağı açıklanır.
- b) Dizi elemanlarını sıralama ve filtreleme işlemleri açıklanır.

**2.9. Javascript programlama dilindeki nesne ve sınıf kavramlarını tanımlar.**

- a) Javascript programlama dilindeki nesnelerin ve sınıfların tanımlanması açıklanır.
- b) Nesnelerin özellikleri ve metotlarının kullanılma şekli açıklanır.

**2.10. Constructor fonksiyonuyla nesne oluşturmeyi kavrar.**

- a) Constructor fonksiyonlarının tanımlanarak nesne oluşturulması açıklanır.
- b) Constructor fonksiyonları ile nesnelere özellik ekleme durumu açıklanır.
- c) Constructor fonksiyonları ile nesnelere özelleştirme durumu açıklanır.

**2.11. Nesnelere özellik ve metotlar ekleyerek diğers nesnelere etkileşim kurmayı kavrar.**

- Nesneler arası iletişim ve etkileşim sağlama yöntemleri açıklanır.
- Nesneler arasında veri paylaşma ve metot çağırma işlemleri yapılır.

### Değerler

Sabır (Kazanım 2.1., 2.2., 2.3., 2.5., 2.7.)

Özdenetim (Kazanım 2.2., 2.3., 2.5., 2.7., 2.10.)

Sorumluluk (Kazanım 2.2., 2.3., 2.4., 2.7., 2.8., 2.9., 2.11., 2.12.)

Yardımselverlik (Kazanım 2.2., 2.3., 2.6., 2.7., 2.8., 2.9., 2.12.)

### Alan Becerileri

Soyutlama Becerisi (Kazanım 2.1., 2.2., 2.3., 2.5.)

Mantıksal Düşünme Becerisi (Kazanım 2.2., 2.3., 2.4., 2.5., 2.6., 2.10., 2.11.)

Algoritmik Düşünme Becerisi (Kazanım 2.2., 2.3., 2.4., 2.5., 2.6., 2.7., 2.8., 2.9., 2.10., 2.11.)

Kodlama, Programlama Becerisi (Kazanım 2.1., 2.2., 2.3., 2.4., 2.5., 2.6., 2.7., 2.8., 2.9., 2.10., 2.11., 2.12.)

Genelleme Becerisi (Kazanım 2.11., 2.12.)

## 3. ÜNİTE: JAVASCRIPT PROGRAMLAMA DİLİ İLE WEB GELİŞTİRME

### Ünite Açıklaması

Bu ünite; Javascript programlama dilinde DOM kavramı, DOM'un web sitelerindeki rolü, DOM aracılığı ile HTML elemanlarına erişimden bahsedilir. Javascript programlama dilinde eventler kullanılarak kullanıcı ile etkileşim kurma, asenkron programlamanın web sitelerindeki önemi üzerinde durulur. Jquery ve Ajax kütüphaneleri ile etkileşim kurma, fetch işlemleri ile asenkron veri alışverişi yapma işlemleri anlatılır. Javascript programlama dili kodlarını modüler hâle getirerek kullanmanın öneminden, NodeJS ve CommonJS modülleri kullanılarak arkaplanda yapılabilecek işlemlerden, NPM aracılığı ile Javascript programlama dili içerisine harici kütüphaneler yüklenebileceğinden bahsedilir. Javascript programlama dilinin popüler framework'lerinden React, Angular ve Vue kullanılarak web siteleri oluşturabilme, React Native framework'ü ile mobil uygulamalar oluşturabilme işlemleri anlatılacaktır. Ecmascript kullanarak Javascript programlama dilinin güncel standartlarına uygun şekilde çalışılabileceğinden bahsedilecektir.

### Kazanım ve Açıklamaları

#### 3.1. DOM kavramını tanımlar.

- DOM (Document Object Model) teriminin ne anlama geldiği açıklanır.
- Web sayfalarının ağaç yapısı olarak temsil edilmesi ve bu yapının kullanılma şekli açıklanır.
- DOM ağacının her düğümünün birer HTML elemanını temsil etmesi vurgulanır.

#### 3.2. DOM'un web sayfalarındaki rolünü kavrar.

- DOM'un web sayfalarındaki içeriği, yapısı ve görünümü açıklanır.

b) Kullanıcı arayüzü ile etkileşim kurma, içerik ekleme veya değiştirme işlemlerinde DOM'un önemi vurgulanır.

c) DOM'un web geliştirme sürecinde aldığı rol vurgulanır.

### 3.3. DOM aracılığı ile HTML elemanlarına nasıl erişileceğini kavrar.

a) DOM üzerinden HTML elemanlarına erişme yöntemi açıklanır.

b) Elemanlar seçilerek üzerinde değişiklik yapma yöntemleri açıklanır.

### 3.4. Kullanıcılar ile etkileşim kurmayı kavrar.

a) Kullanıcı etkileşimi oluşturmak için Javascript programlama dili eventlerinin (olaylarının) kullanılma şekli açıklanır.

b) Event dinleyicileri eklenerek sayfa üzerinde tepki verme işlemleri açıklanır.

### 3.5. Asenkron programlamanın web uygulamalarındaki önemini kavrar.

a) Asenkron programlama açıklanarak bunun web uygulamalarındaki önemi vurgulanır.

b) Asenkron programlamanın sayfa hızı ve kullanıcı deneyimine sağladığı katkı vurgulanır.

c) Callback fonksiyonlar ve async/await gibi asenkron programlama teknikleri açıklanır.

### 3.6. JQuery ve Ajax kütüphaneleri ile etkileşimli şekilde çalışmayı kavrar.

a) JQuery ve Ajax kavramları açıklanır.

b) JQuery ile HTML elemanlarının manipüle edilme yöntemleri açıklanır.

c) Ajax kullanılarak sunucu ile veri alışverişi yapma işlemi açıklanır.

### 3.7. Fetch işlemlerini kullanarak asenkron çalışmayı kavrar.

a) Fetch API kullanılarak sunucudan verilerin nasıl alınabileceği açıklanır.

b) JSON formatında çekilen verilerin nasıl işleneceği ve kullanılacağı açıklanır.

c) Fetch işlemlerinin web uygulamalarındaki kullanılma şekilleri örneklerle açıklanır.

### 3.8. Javascript programlama dili kodlarını daha modüler hâle getirerek kullanmanın önemini kavrar.

a) Kodu parçalara bölerek modüler çalışmanın önemi vurgulanır.

b) Javascript programlama dili modüllerini oluşturup kullanma konusunda örnek çalışmalar yapılır.

### 3.9. NodeJS modülünü kullanarak arkaplanda işlemler yapılabilceğini kavrar.

a) NodeJS ile sunucu tarafında Javascript programlama dilinin geliştirilme işlemi açıklanır.

b) NodeJS ile dosya ve ağ işlemleri gibi arkaplan işlemlerinin gerçekleştirilebileceği vurgulanır.

### 3.10. CommonJS modülünü kullanarak arkaplanda işlemler yapılabilceğini kavrar.

CommonJS modül sistemi kullanılarak dosyaların modüler hâle getirilebileceği vurgulanır.

### 3.11. Npm aracılığı ile harici kütüphaneler kullanarak modül ekleme işlemini kavrar.

a) Npm (Node Package Manager) aracılığıyla harici Javascript kütüphanelerinin projeye eklenme şekli



açıklanır.

b) Harici kütüphanelerin projeye dahil edilmesi işlemleri açıklanır.

c) Harici kütüphanelerin proje içerisinde güncellenmesi gibi yönetim işlemleri açıklanır.

**3.12. Javascript programlama dilindeki yaygın framework'lerin web geliştirmedeki rolünü kavrar.**

a) Yaygın Javascript programlama dili framework'leri açıklanır ve bu framework'lerin önemi vurgulanır.

b) Framework'lerin kullanılma şekli ve web uygulamalarını geliştirirken üstlendiği rol açıklanır.

c) Framework'lerin avantaj ve dezavantajları belirtilir.

**3.13. Farklı açık kaynak kodlu web geliştirme framework'lerinin mobil uygulama geliştirmede nasıl kullanılacağını kavrar.**

a) Açık kaynak kodlu mobil uygulama geliştirme frameworklerinin ne olduğu ve nasıl çalıştığı açıklanır.

b) Web geliştirme bilgisinin mobil uygulama geliştirmede kullanılan framework'lere uyarlanarak mobil uygulama geliştirme sürecine sağladığı katkı vurgulanır.

c) Mobil uygulama geliştirme süreçleri açıklanır.

**3.14. Gelişmekte olan EcmaScript ile Javascript programlama dilinin güncel standartlara uyumlu şekilde çalışabileceğini kavrar.**

a) EcmaScript'in Javascript programlama dilinin güncelleme ve yeni özelliklerini içermesi durumu açıklanır.

b) Javascript programlama dilindeki kodun güncel standartlara uyumlu hâle getirmenin önemi vurgulanır.

**Değerler**

Sabır (Kazanım 3.1., 3.2., 3.3., 3.5.)

Özdenetim (Kazanım 3.4., 3.5., 3.6., 3.7., 3.14.)

Sorumluluk (Kazanım 3.4., 3.5., 3.7., 3.14.)

Yardımselik (Kazanım 3.11., 3.12., 3.13.)

**Alan Becerileri**

Soyutlama Becerisi (Kazanım 3.1., 3.2., 3.3.)

Mantıksal Düşünme Becerisi (Kazanım 3.4., 3.5., 3.6., 3.7.)

Algoritmik Düşünme Becerisi (Kazanım 3.4., 3.5., 3.6., 3.7.)

Kodlama, Programlama Becerisi (Kazanım 3.4., 3.5., 3.6., 3.7., 3.8., 3.9., 3.10., 3.11.)

Genelleme Becerisi (Kazanım 3.11., 3.12., 3.13., 3.14.)

Test Etme ve Hata Ayıklama Becerisi (Kazanım 3.14.)

## PROGRAMLAMA DİLLERİ (Python)

Ön koşul olan programlamaya giriş becerileri üzerine inşa edilen Python programlama dili, öğrencilerin temel programlama becerilerini bir programlama diline uyarlayarak geliştirmelerini amaçlamaktadır. Bu beceriler ile öğrencilere hayatlarının tüm aşamalarında kullanacakları öğrenmeyi öğretme ve üreten bir nesil olmaları için gereken becerilerin kazandırılması hedeflenmiştir. Ders içeriği, öğrencilerin sorgulama, aktif katılım, özgün düşünme ve pratiğe dayalı öğrenme yoluyla içsel motivasyonlarını artırarak özgüvenlerini geliştirmeyi amaçlamaktadır. Bilgi işlemsel düşünme ve programlama becerileri öğrencilerin teknoloji ile bütünleşmesini sağlayacaktır. Bilgi işlem cihazları ile aynı dili konuşabilmek, onları yöneten ya da onlar tarafından yönetilen bir birey olmanın temel farkını ortaya koyacaktır. Bu bağlamda, öğrencilerin teknoloji ve kodlama okuryazarlığına sahip üreten ve teknoloji yöneten bir gelecek oluşturmasında algoritma tasarlama ve bilgi işlemsel düşünme becerileri temelli bir programlama dersi almaları planlanmaktadır.

Python programlama dili dersi ile öğrencilerin aşağıda belirlenen hedeflere ulaşmaları beklenmektedir:

- Bilgi iletişim teknolojilerini etkili ve amacına uygun kullanmaları
- Python programlama dili ile programlama yazılım geliştirmeye ilişkin genel bir anlayış ve teknik birikime sahip programlama okuryazarı olmaları
- Algoritma tasarımına ilişkin anlayış geliştirerek sözel ve görsel olarak ifade edebilmeleri
- Öğrendiği bilgileri farklı durumlara aktarabilmeleri
- Python programlama geliştiricilerinden (IDE) en az birini iyi düzeyde kullanabilmeleri
- Uygulama geliştirirken hangi tasarım ilkelerine dikkat etmesi gerektiğini bilmeleri
- İnternet ortamında kişisel verilerin korunmasının gerekliliğini bilmeleri ve bilgi güvenliğini sağlamaya yönelik önlemler alabilmeleri
- Günlük hayatta karşılaşılan sorunların (yaşlı ve engelli bireylerin karşılaştığı sorunlar, çevre, ulaşım sorunları vb.) çözümüne ilişkin yenilikçi ve özgün projeler geliştirebilmeleri
- Öğrenme sürecinde iş birlikli çalışma becerisine sahip, sosyal ortamlardan faydalanabilen bireyler olmaları
- Yaşam boyu öğrenmeye istekli bireyler hâline gelmeleri
- Python programlama dilinde nesne yönelimli programlama teknikleri ile proje geliştirmeleri, geliştirdiği projede form uygulamalarına yer vermeleri ve geliştirdiği projenin bir veri tabanı ile bağlantısını kurabilmeleri

### 1. ÜNİTE: PYTHON PROGRAMLAMAYA GİRİŞ

#### Ünite Açıklaması

Bu üniteye sırasıyla; Python çalışma ortamı türleri, değişken tanımlama, veri türleri, tanımlanan değişkenlerde işlem önceliği ve kullanıcıdan alınan verinin değişkene aktarılması üzerinde durulur.

#### Kazanım ve Açıklamaları

##### 1.1. Python programlama diliyle çalışılacak ortamları tanıy.

- a) IDE (Integrated Development Environment) kavramı açıklanır.
- b) Python IDE türlerini listeler. Öğrencilere popüler IDE'lerin avantaj ve dezavantajları açıklanır.
- c) Derlenen ve yorumlanan dillerden ve bunların farklarından bahsedilir.
- ç) Derleyici ve yorumlayıcı örnekleri verilir. Derleyici ve yorumlayıcı kullanmanın avantaj ve dezavantajları anlatılır.
- d) Python programlama dilinin yorumlanan bir dil olduğu vurgulanır.
- e) En yaygın kullanılan IDE'lerden (Atom, IDLE, Jupyter, PyCharm, Pydev, Spyder, Sublime Text ve Visual Studio Code) bahsedilir.

**1.2. Farklı entegre geliştirme ortamlarını analiz ederek kendisine en uygun çalışma ortamını seçer.**

- a) Ders akışında kullanılacak Python ide seçilir, kurulumu yapılır. Seçilen Python idesi öğrencilere tanıtılır. Python programlama diline özgü basit örnekler ile editörde üretilen sonuç gözlemlenir. Kod satırına yazılan 5+2 ifadesinden elde edilen sonuç ile Python programlamanın işlevselliği ve kodlama kolaylığı hususunda idenin rolü açıklanır.
- b) Özellikleri anlatılan bir Python IDE'si seçilerek öğrencilerle birlikte kurulumu yapılır.
- c) Hangi satırların hangi renklerde görüldüğü örneklerle açıklanır.
- ç) Yorum satırından bahsedilir. Tek satır ve çok satır olması durumunda kullanılacak ifadeler gösterilir.
- d) Ekran yazdırma komutu print()'in önemi ve gerekliliği vurgulanır.
- e) Kullanıcıdan veri alınması gereken durumlarda input() komutunun kullanıldığı örneklerle gösterilir.
- f) Farklı Python çalışma ortamlarının avantaj ve dezavantajları listelenir.
- g) Öğrencilerden kullanacakları Python çalışma ortamını, nedenlerini belirterek seçmeleri istenir.

**1.3. Değişkenlerde tutulacak verileri tanımlar.**

- a) Python programlama dilinde kullanılan veri türleri listelenir. Metinsel, sayısal, sıralama (list, tuple) ve sözlük türünde veri tanımlamaları yapılır.
- b) Değişken tanımlama kurallarından bahsedilerek verilerin değişkenlere ataması yapılır.
- c) Değişken ve sabit kavramlarının tanımı yapılır.
- ç) Bir değişkene veri ataması örnekler üzerinden anlatılır.
- d) Değişken isimleri belirlerken dikkat edilmesi gereken kurallar açıklanır, kullanılabilecek bazı standartlar (camel case, pascal base, vb.) tanıtılır.

**1.4 Bir değişkeni tanımlarken hangi tür veriyi tutacağını belirler.**

- a) Değişkenlerin farklı türlerde verileri (tam sayı, ondalıklı sayı, metin, vb.) tutabileceği vurgulanır.
- b) Veri türlerinden bahsedilir. Kullanım yerleri örneklerle açıklanır.
- c) Kullanım amacına göre veri türü dönüşümlerinden bahsedilir.

**1.5. Tanımlanan değişkenlerde işlem önceliğini gösterir.**

- a) Programda kullanılan operatörler listelenir. Aritmetiksel, mantıksal ve atama operatörleri örneklendirilir.

Farklı değişkenlerdeki verilerle karşılaştırma operatörlerini kullanarak işlemler yapılır.

- b) Operatörlerle yapılan işlemlerde işlem önceliği gözetilir, farklı sonuçlar elde edilebileceği durumlar örneklendirilir.
- c) Operatörler ile aritmetiksel ve mantıksal işlemlerden bahsedilir.
- ç) Eşitlik ve atama operatörü arasındaki fark örneklerle gösterilir.
- d) Matematik dersindeki işlemler ile benzerlikleri anlatılır ve işlem sıralaması bir tablo ile gösterilebilir.

#### **1.6. Kullanıcıdan veri alır.**

- a) Python'da kullanıcıdan veri almak için kullanılan "input" komutunun kullanımı açıklanır.
- b) Kullanıcıdan veri almak için input() komutu kullanılacağı gösterilir.

#### **1.7. Kullanıcıdan alınan veriyi değişkene tanımlar.**

- a) Alınan verinin değişkene atılması örneklerle gösterilir.
- b) input() komutunu kullanarak aldığı veriyi değişkene atar.
- c) Verinin kullanım amacına göre türünün değiştirilmesi gerektiğinden bahsedilir. Bu değişimin nasıl olacağı örneklerde gösterilir. Örneğin matematiksel bir işlem yapılacaksa kullanıcıdan alınan verinin integer türüne dönüştürülmesi için input() komutunun başına int() ifadesinin yazılması gerektiği söylenir.

#### **Değerler**

Öz denetim (Kazanım 1.1., 1.2.)

Sorumluluk (Kazanım 1.1., 1.2., 1.3., 1.4., 1.5.)

#### **Alan Becerileri**

Veri Toplama (Kazanım 1.3., 1.4., 1.5.)

Veri Gösterimi ve Analizi (Kazanım 1.3., 1.4., 1.5.)

Kodlama, Programlama Becerisi (Kazanım 1.1., 1.2., 1.3., 1.4., 1.5.)

Teknik İşlem Becerisi (Kazanım 1.1., 1.2.)

## **2. ÜNİTE: PYTHON PROGRAMLAMA DİLİ İLE YAZILIM GELİŞTİRME**

### **Ünite Açıklaması**

Bu üniteye sırasıyla; karar yapılarını kullanma, döngüleri kullanarak bir problem çözme, çoklu koşulların mantıksal sonuçlarını açıklama, fonksiyonları program içinde yöntemine göre kullanma ve standart fonksiyonlar ve kütüphaneleri üzerinde durulur.

### **Kazanım ve Açıklamaları**

#### **2.1. Python programlama dili karar yapılarını içeren programlar geliştirir.**

- a) Karar yapıları günlük hayattan örneklerle açıklanır. Çoklu karar durumlarında nasıl bir sıralama izlendiği

belirtilir. If-else ve if-elif-else koşul durumları karşılaştırılır, farklı problem durumlarıyla kullanılır.

- b) İç içe karar yapıları açıklanır. Örnek programlar yazılır.
- c) If else yapısının kullanımı açıklanır. Bu yapı kullanılarak problem çözmesi istenir.
- ç) Elif yapısının kullanımı açıklanır. Bu yapı kullanılarak problem çözmesi istenir.
- d) Boolean yapısı anlatılır.
- e) If, if else, if elif else yapılarının kullanımı açıklanır.
- f) Karar yapısı içeren farklı problemleri belirtilen yapılar kullanarak çözecek kodların yazılması istenir.

### **2.2. Bir problemi çözmek için kullanacağı döngü yapısına karar verir.**

- a) Tekrar eden durumları programlarken for, while gibi döngüleri amacına uygun olarak kullanır.
- b) İç içe döngü yapıları kullanımı açıklanır. Örnek programlar yazılır.
- c) While ve for döngülerinin program içinde nasıl kullanıldığı anlatılır.
- ç) While ve for arasındaki farktan bahsedilir. Hangi durumlarda kullanılacağına dair örnekler verilir.
- d) Döngünün belirli bir aralıkta çalışmasını sağlayan range kavramı örneklerle açıklanır.
- e) Break ve continue komutlarına örnek verilir.

### **2.3. Çoklu koşulların mantıksal sonuçlarını değerlendirir.**

- a) Çoklu koşulların kullanımında mantıksal operatörlerin doğurabileceği sonuçlar tartışılır. Pozitif, negatif mantık yapısı açıklanır. Koşul durumlarında tek tip mantık yapısı üzerinden gidilmesi tavsiye edilir.
- b) İç içe döngülerden bahsedilir.
- c) Öğrencilerden iç içe döngüleri kullanılarak çarpım tablosu yapmaları istenir. Örnek üzerinden mantıksal sonuç çıkarımı için sınıfça tartışılır.
- ç) Öğrencilere içinde tekrar eden komutların bulunduğu kod parçaları verilir. Bu kodların aynı komutların tekrarı olmadan yazılıp yazılamayacağı üzerinde düşünmeleri istenir.
- d) For/while döngü yapılarının kullanımı açıklanır.
- e) Döngü kullanımının kod tekrarını önleyerek programı sadeleştirdiği vurgulanır.

### **2.4. Fonksiyonları program içinde yöntemlerine göre kullanır.**

- a) Programlama dilinde fonksiyon kullanımı açıklanır. Fonksiyon tanımlaması yapılıp, tanımlanan fonksiyonun çağrılması durumunda gerçekleştireceği işlemler tanımlanır. Parametrelili ve parametresiz fonksiyonlar yazılarak farkı tartışılır.
- b) Fonksiyon kavramı ve kullanım gerekçeleri açıklanır.
- c) Bir programda farklı yerlerde aynı işlemi yapmak için kodları kopyalayıp yapıştırmak yerine fonksiyonların da kullanılabileceği örneklerle gösterilir.
- ç) Öğrencilerden kendi girdikleri ismi, ekrana yazdırma işlemi yapan bir fonksiyon tanımlamaları istenir.
- d) Parametrelili ve parametresiz fonksiyon arasındaki fark örneklerle açıklanır.

**2.5. Standart fonksiyon kümelerini ve kütüphanelerini kullanır.**

- a) Ön tanımlı fonksiyon kümelerine yer verilir. Liste üzerinde fonksiyon işlemleri yapılır. Len (), count (), insert () vb. hazır fonksiyonlar örneklendirilir.
- b) Math() kütüphanesi kullanılır. math.factorial (8) işleminin sonucu gözlemlenir. Çeşitli hazır fonksiyon kütüphaneleri öğrencilere tanıtılır.
- c) Öğrencilerden Python programlama dilinde sıklıkla kullanılan ve önemli işlemlere sahip standart fonksiyonlar ile ilgili araştırma yapmaları istenir.
- ç) Gömülü fonksiyonların yanında hazır kütüphanelerin varlığından bahsedilir ve örneklendirilerek açıklanır.
- d) Üst düzey fonksiyonlardan bahsedilir.
- e) Kendini çağıran fonksiyonlar örneklendirilir.
- f) Yazılmak istenen her programda kullanıcının daha önce tanımladığı fonksiyonları yeniden tekrarlamak zorunda olup olmadığı sorulur, öğrencilerin fikir yürütmesi istenir.
- g) Python'da yazılan tüm komutların aslında önceden tanımlanmış fonksiyonlar olduğu vurgulanır.
- h) Kütüphane tanımı yapılır, Python'da yaygın kullanılan kütüphanelere örnekler verilir.

**Değerler**

Adalet (Kazanım 2.1., 2.2., 2.3., 2.4., 2.5.)

Özdenetim (Kazanım 2.1., 2.2., 2.3., 2.4., 2.5.)

Sorumluluk (Kazanım 2.1., 2.2., 2.3., 2.4., 2.5.)

Sabır (Kazanım 2.1., 2.2., 2.3., 2.4., 2.5.)

**Alan Becerileri**

Kontrol Yapıları (Kazanım 2.1., 2.2., 2.3.)

Test ve Doğrulama (Kazanım 2.4., 2.5.)

**3. ÜNİTE: NESNE YÖNELİMLİ PROJE GELİŞTİRME****Ünite Açıklaması**

Nesne yönelimli programlama, programlama açısından düşünüldüğünde önemli bir kavramdır. Nesne yönelimli programlama ileri düzey ve gelişmiş programlar yazmayı sağlar. Karmaşık projelerdeki kodların daha iyi organize edilmesi, kod tekrarlarından kurtulma, hata düzeltmeyi kolaylaştırma gibi avantajlar sunar.

Bu üniteye sırasıyla; nesne yönelimli programlamanın faydaları, sınıf tanımlaması, Python'a özgü temel sınıflar, programda kapsülleme, kalıtım ve çok biçimlilik kullanımı, liste, demet, küme, sözlük türündeki veriler, dosya işlemleri ve ileri seviye nesne yönelimli tekniklerle proje geliştirme üzerinde durulur.

**Kazanım ve Açıklamaları**

**3.1. Nesne yönelimli programlama kavramını açıklar.**

- a) Nesne yönelimli programlamanın ve nesne kavramının tanımı yapılır. Attribute (nitelik) ve metot kavramları üzerinde durulur.
- b) Nesnelerin metotlar ve niteliklerden oluştuğu vurgulanır. Niteliklerin nesnelerin sahip oldukları verilere, metotların ise bunlar üzerinde yapılabilecek işlemlere karşılık geldiği açıklanır.
- c) Nesne yönelimli programlama ile yordamsal programlama arasındaki farkı kavrar.
- ç) Nesne kullanımının gerektirdiği durumları açıklar.
- d) Dilin kurallarına uygun nesne tanımlamasını öğrenir.
- e) Dosya ve grafik nesnelere bahsedilir, hangi projelerde bu nesnelere ihtiyaç duyulabileceği üzerine akıl yürütmesi istenir.

**3.2. Nesne yönelimli programlamanın faydalarının farkına varır.**

- a) Nesne yönelimli programlamaya neden ihtiyaç duyulabileceği öğrencilere açıklanır. İleri düzey programlar yazılmak istendiğinde temel programlama kavramlarının yetersiz kalacağına altı çizilir.
- b) Aynı sınıfa tanımlanan verilerde değişiklik yapmak istendiğinde programcıya ne gibi kolaylıklar sağlayacağı üzerine akıl yürütmesi istenir.
- c) Hangi durumlarda nesne tabanlı programlamaya ihtiyaç duyulabileceği örnek projeler üzerinden gösterilir.

**3.3. Programın ihtiyacına yönelik sınıf tanımlaması yapar.**

- a) Farklı niteliklere sahip sınıf tanımlaması yapılır. Tanımlanan sınıfın niteliklerine özgü metotlar yazılır.
- b) Tanımlanan sınıfa birden fazla metot eklenir. Tanımlanan sınıfa özgü nesne oluşturulması sağlanır.
- c) Sınıf oluşturma birkaç örnekle öğrencilere anlatılır.
- ç) Sınıftan nesne tanımlama ve nesnenin özelliklerine değer atama işlemleri hakkında örnekler yapılır.
- d) Öğrencilerden kendi sınıflarını oluşturarak bu sınıflar içinde nesnelere oluşturacakları bir örnek çalışma yapmaları istenir.

**3.4. Python programlama diline özgü temel sınıfları kullanır.**

- a) Python programlama dilinin açık kaynak yapısından dolayı geniş bir kütüphaneye sahip olduğundan bahsedilir. Dolayısıyla her sınıfın yeniden tanımlanması gerekmez önceden tanımlanmış sınıflar da kullanılabilir. Time ve random gibi en çok kullanılan sınıflardan örnekler verilir.
- b) Python programlama diline özgü sınıflar program içerisinde kullanılır.
- c) Modül kavramından bahsedilir. Modüle ait komutlar program içerisinde kullanılır.

**3.5. Geliştirdiği programda kapsülleme (encapsulation), kalıtım (inheritance) ve çok biçimlilik (polymorphism) prensiplerini kullanır.**

- a) Nesne yönelimli programlama teknikleriyle sınıf tanımlaması yapar.
- b) Kapsülleme, kalıtım, çok biçimlilik vb. ileri düzey nesne yönelimli programlama tekniklerini kullanarak program yazar.

**3.6. List, tuple, set ve dictionary türlerinde verileri tanımlar.**

- a) List (liste) veri türü açıklanarak örnek veri tanımlaması yapılır. List içerisindeki öğelerin esnek yapıda olabileceği vurgulanır.
- b) Tuple (demet) veri türü List yapısı arasındaki benzerliklerden bahsedilir. Tuple'da sadece verilerin okunabildiği, veri ekleme veya çıkarma yapılamayacağı açıklanır. Bununla ilgili öğrencilerin bir örnek yazması istenir.
- c) Set (küme) veri türünün tekrarlanabilir, değiştirilebilir ve yinelenen öğeleri olmayan sırasız bir veri türleri koleksiyonu olduğu açıklanır ve örneklerle pekiştirilir.
- ç) Dictionary (sözlük) veri türü; içerisinde yer alan verileri, anahtar kavram ve bu anahtar kavrama karşılık gelen değerler olarak tutar. Öğrencilerden bununla ilgili bir örnek yazması istenir.

**3.7. Dosya nesnesi üzerinde okuma yazma silme işlemlerini uygular.**

- a) Dosya nesnelere kullanmanın programda veri depolama üzerine etkisi tartışılır. Metin dosyalarında var olan veriyi okuma, yazma ve silme işlemleri örneklendirilir.
- b) Verileri kaybetmeme ve daha sonra tekrar erişebilmek için dosyaların önemli olduğu vurgulanır.
- c) Dosya açma, bir dosyayı okuma ve dosyaya yazma işlemlerinin nasıl yapıldığı açıklanır. Dosya nesnesi ile ilgili metotlar üzerinde durulur. Dosya oluşturma ile ilgili örnek kodlama çalışmaları yapılmaları sağlanır.

**3.8. Dizi türündeki verileri dosya nesnesi içerisinde örneklendirir.**

- a) Listede tutulan veriyi dosyaya yazdıran programı kodlar.
- b) Dosyadan okunan veriyi bir listeye tanımlayan programı kodlar.
- c) Dosyalara özgü metotları (with as, tell, seek) kullanır.
- ç) Programda işlevi tamamlanan dosyayı siler.
- d) Dizi türündeki verilerin dosya içerisinde nasıl yazıldığına örneği yapılıdır.
- e) Dizi kavramı açıklanır. Dizin özellikleri ve metotları üzerinde durulur.
- f) Metinsel ifadelerin dizi nesnesi olduğu vurgulanır. Metinsel ifadeler içeren bir dosya oluşturmaları ve bu metinsel ifadelerde metotlarla işlem yapacakları örnek çalışmalar oluşturmaları istenir.

**3.9. İleri seviye nesne yönelimi teknikleriyle özgün projeler geliştirir.**

- a) Sınıf ve nesnelere kullanarak özgün bir proje tasarlar. Geliştirdiği projede sınıfa özgü teknikleri kullanır. Dosya nesnelereyle işlemler yapar.
- b) Projede karşılaşılabileceği olası hataları göze alarak hata ayıklama yöntemlerini uygular.
- c) Python için tanımlanan hazır grafik arayüzlerden bahsedilir. Bunlar kullanılarak özgün bir proje geliştirilmesi istenir.
- ç) Öğrencilerden özgün bir proje fikri sunmaları istenir. Proje fikirleri ile ilgili ön analiz çalışmaları yapılmaları konusunda yönlendirilir.
- d) Nesne yönelimli programlama özelliklerini kullanarak proje fikirlerini hayata geçirmeleri beklenir. Süreç boyunca öğretmen tarafından öğrencilere gerekli destek sağlanmalıdır. Proje sonunda öğrencilerden



bir sunum gerçekleştirerek projelerini tanıtmaları, süreç içerisinde zorlandıkları ve gerçekleştiremedikleri noktaları paylaşmaları istenir.

### Değerler

Yardımselik (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)

Sabır (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6., 3.7., 3.8., 3.9)

Adalet (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6., 3.7., 3.8.)

### Alan Becerileri

Veri Toplama (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)

Veri Gösterimi ve Analizi (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)

Model Doğrulama (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)

Test ve Doğrulama (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)

Kontrol Yapıları (Kazanım 3.1., 3.2., 3.3., 3.4., 3.5., 3.6.)

Algoritmik Düşünme Becerisi (Kazanım 3.8.)

Kodlama, Programlama Becerisi (Kazanım 3.2.,3.6.,3.7.,3.8.)

## YAPAY ZEKÂ UYGULAMALARI

Bu derste yapay zekâ uygulamaları gerçekleştirmek için öğrencilerin veriyi elde etme, temizleme, dönüştürme, analiz etme ve görselleştirme basamaklarını kavramaları sağlanmaktadır. Öğrenciler Python programlama dili kütüphanelerini ve algoritmaları kullanarak veriye ait bir senaryo oluşturacak ve istatistiksel modeller kuracaklardır. Ders süresince öğrenciler veri bilimcilerin anlamlı içgörüler elde etmek ve organizasyonel sorunları çözmek için kullandıkları araçları kullanacaklardır. Öğrenciler yapay zekânın nasıl kullanıldığını ve topluma yansıyan sosyal ve etik etkilerini öğreneceklerdir. Doğrusal ve lojistik regresyon kullanarak tahmine dayalı modeller yapmayı öğreneceklerdir. Son olarak bir sohbet robotu oluşturacaklardır.

### 1. ÜNİTE: VERİ BİLİMİ TEMEL KAVRAMLARI

#### Ünite Açıklaması

Bu ünite de öğrenciler veri biliminin temel kavramlarını öğrenir. Veri bilimi alanında çalışmak için Python programlama dilinin önemini kavrar ve çalışma ortamını hazırlar. Veri etiği konusunda bilgi sahibi olur. Python programlama dili altında kullanılan temel modüller, paketler ve kütüphaneleri kullanır.

#### Kazanım ve Açıklamaları

##### 1.1. Veri bilimi temel kavramlarını açıklar.

- a) Veri biliminin temel kavramları açıklanır.
- b) Veri biliminin kullanıldığı alanlar ve günlük hayata olan etkisi açıklanır.

##### 1.2. Soru tiplerini açıklar.

- a) Doğru verilere ulaşmak için doğru soru sormanın önemi açıklanır.
- b) Veriye dayalı olarak karar verme sürecinin önemi vurgulanır.

##### 1.3. Veri bilimcilerin çalışma basamaklarını açıklar.

- a) Veri bilimcinin çalışma alanları açıklanır.
- b) Veri bilimcide bulunması gereken özellikler açıklanır.

##### 1.4. Keşifçi veri analizini açıklar.

- a) Veriyi her yönüyle keşfetmenin veri analizindeki önemi vurgulanır.
- b) Veriler arasındaki ilişkinin ve veriyi görselleştirmenin veri analizi sürecine etkilerinden bahsedilir.

##### 1.5. Veri döngüsü basamaklarını açıklar.

- a) Veri döngüsü basamakları açıklanır.
- b) Verilerin analiz edilmesi, modelleme yapılması, sonuçların yorumlanması ve tekrardan kullanılabilmesi için veri döngüsüne olan ihtiyaç vurgulanır.

##### 1.6. Veri türlerini açıklar.

- a) Nicel ve nitel veri kavramları açıklanır.
- b) Veri yönetiminin kolaylaşması için veri türlerinin önemi vurgulanır.

**1.7. Veri etiği kavramını açıklar.**

- Veri etiği, veri mahremiyeti, dijital veri kavramları açıklanır.
- Günlük hayatta kullanılan verilerin mahremiyeti ve veri etiği üzerindeki etkisi açıklanır.
- Kişisel verilerin korunması ve veri mahremiyetinin sağlanması için bireylere ve toplumlara düşen görevler açıklanır.

**1.8. Python programlama dili için etkileşimli tasarım ortamı kurar.**

- Python programlama dilinin kullanıldığı öğrenme ve çalışma ortamı hazırlanır.
- Veri analizinde yaygın olarak kullanılan etkileşimli tasarım ortamları üzerinde durulur.

**1.9. Python programlama dili temellerini veri bilimi bağlamında kullanır.**

- Python programlama dilinde mantıksal operatör kavramı açıklanır.
- Mantıksal operatörlerle ilgili uygulamalar yapılır.
- Python programlama dilinde liste ve değişken kavramı açıklanır.
- Liste ve değişkenlerle ilgili uygulama yapılır.

**1.10. Açık kaynak kodlu programlama dili kavramını tanımlar.**

- Açık kaynak kodlu programlama dili kavramı açıklanır.
- Açık kaynak kodlu programlama dili çalışmalarında geliştirici topluluklarının önemine değinilir.

**1.11. Veri bilimi için gerekli olan Python programlama dili kütüphanelerini tanımlar.**

Çeşitli kütüphaneler (Numpy, Pandas, Scikit-learn, Matplotlib veya Seaborn vb.) tanıtılır.

**1.12. Array veri yapısını kullanır.**

- Array veri yapısı kavramı tanımlanır ve örnekler verilir.
- Array veri yapısı ile liste veri yapısının kullanımı karşılaştırılır.
- Farklı metotlar kullanarak array oluşturulur.
- Array veri yapısı içerisindeki elemanları bulma işlemleri gerçekleştirilir.

**1.13. Arraylerde birleştirme, sıralama, ayırma ve matematiksel işlemler yapar.**

- Herhangi bir array kullanılarak farklı arrayler oluşturulur.
- Arraylerde birleştirme, sıralama, ayırma ve matematiksel işlemler gerçekleştirilir.

**1.14. Arraylerde biçim, boyut ve boy işlemlerini uygular.**

- Arraylerde şekil, boyut ve boy kavramları açıklanır.
- Arraylerde şekil, boyut ve boy bilgilerine ulaşmak için shape, reshape, ndim ve size komutlarının kullanımı gösterilir.
- Bir arraydeki değişkenlerin tipini belirlemek için dtype komutunun kullanımı gösterilir.

**1.15. Veri seti oluşturarak veri çerçevesine dönüşüm işlemleri gerçekleştirir.**

- Liste, array ve sözlük veri yapıları kullanılarak bir veri seti oluşturulur.
- Pandas kütüphanesi kullanılarak veri seti, veri çerçevesine dönüştürülür.
- Veri seti ile ilgili temel özellikleri incelemek için head(), tail(), columns, axes, values(), types, info() ve describe() komutlarının kullanımı gösterilir.

**1.16. Belirli koşullara dayanarak bir veri çerçevesini filtreler.**

- Veri çerçevelerinde belirli koşulları karşılayan satırları seçmek için koşullu ifadelerin kullanımı gösterilir.
- Veri çerçevesinin belirli sütunlarını seçmek için sütun isimlerin kullanımı gösterilir.
- Belirli değerlere sahip satırları seçmek için isin() fonksiyonunun kullanımı gösterilir.
- SQL benzeri ifadelerle veri çerçevesini filtrelemek için query() fonksiyonunun kullanımı gösterilir.
- Bir sütundaki değerlerin belirli bir aralıkta olup olmadığını kontrol etmek için between() fonksiyonunun kullanımı gösterilir.
- Belirli satır ve sütunları seçmek için loc[] ve iloc[] fonksiyonunun kullanımı gösterilir.

**1.17. Bir veri çerçevesinde indeksleme, satır ve sütun işlemlerini gerçekleştirir.**

- Bir veri çerçevesinin indekslerini ayarlama ve sıfırlama işlemleri gösterilir.
- Bir veri çerçevesi üzerinde fonksiyon kullanımı gösterilir.
- Bir veri çerçevesi üzerinde yeni sütun oluşturma işlemi gösterilir.

**Değerler**

Sabır (Kazanım 1.4., 1.5., 1.6., 1.8., 1.9.)

Sorumluluk (Kazanım 1.1., 1.2., 1.3., 1.10.)

Vatanseverlik (Kazanım 1.7.)

Özdenetim (Kazanım 1.11., 1.12., 1.13., 1.14., 1.15., 1.16., 1.17.)

**Alan Becerileri**

Kodlama, Programlama Becerisi, Veri Toplama, İşleme, Dönüştürme Becerisi (Kazanım 1.12., 1.13., 1.14., 1.15., 1.16.)

Mantıksal Düşünme Becerisi (Kazanım 1.2., 1.3., 1.4., 1.5., 1.6., 1.7., 1.8., 1.9., 1.11., 1.10.)

Genelleme Becerisi (Kazanım 1.9., 1.10., 1.11., 1.12., 1.13., 1.14., 1.15., 1.16., 1.17.)

**2. ÜNİTE: TEMEL İSTATİSTİK****Ünite Açıklaması**

Bu ünite veri biliminde sıklıkla kullanılan istatistiksel yapılardan bahsedilir. Veri analizinde merkezi eğilim ve yayılım ölçülerinin hesaplanması konusu üzerinde durulur.

**Kazanım ve Açıklamaları**

**2.1. Bir seri veri yapısı oluşturur.**

- Bir kütüphane (Pandas) kullanılarak seri veri yapısı oluşturulur.
- Bir seri veri yapısının liste ve sözlük veri yapılarıyla olan farkı açıklanır.

**2.2. Bir serinin ortalama, mod ve medyan değerlerini hesaplar.**

- Merkezi dağılım ölçülerinden ortalama, mod ve medyan kavramları açıklanır.
- Pandas kütüphanesindeki metotlar kullanılarak nümerik değerlerden oluşan bir serinin ortalama, mod ve medyan değerlerinin nasıl bulunacağı gösterilir.

**2.3. Bir serinin merkezi eğilim ve yayılım ölçülerini hesaplar.**

- Merkezi yayılım ölçülerinde standart sapma ve varyans kavramları açıklanır.
- Bir serinin merkezi eğilim ve yayılım ölçüleri hesaplanarak seri hakkında çıkarım yapılır.

**Değerler**

Sabır (Kazanım 2.1., 2.2., 2.3.)

**Alan Becerileri**

Veri Toplama, İşleme Dönüştürme (Kazanım 2.1., 2.2., 2.3.)

**3. ÜNİTE: DEĞİŞİM VE VERİ****Ünite Açıklaması**

Bu ünite de gerçek yaşam verilerinin nasıl yorumlanacağı üzerinde durulur. Bir veriyi çalışma ortamına aktarma, veriyi filtreleme, indeksleme, dilimleme ve birleştirme işlemlerinden bahsedilir.

**Kazanım ve Açıklamaları****3.1. Toplumsal problemler üzerinde veri biliminin etkilerini açıklar.**

- Toplumunu ilgilendiren problemler üzerinde durularak veri biliminin bu problemleri nasıl etkilediği açıklanır.
- Bir sorunu daha iyi anlamak ve insanların bu sorundan nasıl etkilendiğini kavramak için verilerin önemi açıklanır.

**3.2. Veri setleri ile ilgili uygulamalar yapar.**

- Verilerin bilgisayarda csv, json ve excel dosya biçimlerinde saklanabileceğine değinilir.
- Bilgisayar üzerinde kayıtlı olan csv dosyasını çalışma ortamına aktarma işlemi gerçekleştirilir.
- Veri seti içerisinde bulunan belirli sütunlara ait tüm değerleri görmek için filtreleme işlemi gerçekleştirilir.
- Satır numaralarına istenilen değeri vermek için index komutunun kullanımı gösterilir.
- Bir veri setinde bulunan belirli satır ve sütunları silmek için drop() komutuyla beraber axis ve inplace parametrelerinin kullanımı gösterilir.

- e) Bir veri setinde indeks numaraları kullanılarak satır ve sütunlarda bulunan değerlere ulaşma gösterilir.
- f) Bir veri setinde indeksleme ve dilimleme işlemleri için Loc ve iloc komutlarının kullanımı gösterilir.
- g) Veri setlerini birleştirmek için concat() komutu kullanımı gösterilir.

### Değerler

Özdenetim (Kazanım 3.1., 3.2.)

### Alan Becerileri

Kodlama, Programlama Becerisi (Kazanım 3.1., 3.2.)

## 4. ÜNİTE: VERİ GÖRSELLEŞTİRME

### Ünite Açıklaması

Bu ünite de verinin daha iyi anlaşılması, yorumlanması için veri görselleştirmenin öneminden bahsedilir. Çizgi grafiği, histogram grafiği, kutu grafiği, bar grafiği ve saçılma grafiklerinin nasıl oluşturulacağı gösterilir.

### Kazanım ve Açıklamaları

#### 4.1. Veriyi görselleştirmek için çizgi grafiği çizer.

Çizgi grafiği çizmek için plot() fonksiyonunun kullanımı ile ilgili bir uygulama yaptırılır.

#### 4.2. Veriyi görselleştirmek için histogram grafiği çizer.

Histogram grafiği çizmek için hist() fonksiyonunun kullanımı ile ilgili bir uygulama yaptırılır.

#### 4.3. Veriyi görselleştirmek için kutu grafiği çizer.

Kutu grafiği çizmek için box() fonksiyonunun kullanımı ile ilgili bir uygulama yaptırılır.

#### 4.4. Veriyi görselleştirmek için bar grafiği çizer.

Bar grafiği çizmek için bar() fonksiyonunun kullanımı ile ilgili bir uygulama yaptırılır.

#### 4.5. Veriyi görselleştirmek için saçılım grafiği çizer.

Bar grafiği çizmek için scatter() fonksiyonunun kullanımı ile ilgili bir uygulama yaptırılır.

### Değerler

Sabır (Kazanım 4.1., 4.2., 4.3., 4.4., 4.5.)

### Alan Becerileri

Veri Gösterimi ve Analizi (Kazanım 4.1., 4.2., 4.3., 4.4., 4.5.)

## 5. ÜNİTE: VERİLER ARASI İLİŞKİ

### Ünite Açıklaması

Bu ünite de iki veri kümesi arasındaki ilişkinin matematiksel olarak hesaplanması öğretilir.

**Kazanım ve Açıklamaları****5.1. Bağımlı ve bağımsız değişken kavramını tanımlar.**

- Bağımlı değişken kavramı bir örnekle açıklanır.
- Bağımsız değişken kavramı bir örnekle açıklanır.
- Bağımlı değişken ve bağımsız değişken arasındaki fark açıklanır.

**5.2. İki değişken arasındaki ilişkiyi yorumlar.**

- Korelasyon ve nedensellik kavramları açıklanır.
- Korelasyon kullanarak iki değişken arasındaki ilişki açıklanır.

**5.3. Doğrusal regresyon modeli oluşturur.**

- Doğrusal regresyon kavramı açıklanır.
- İki değişken arasındaki ilişkiyi doğrusal bir şekilde açıklamak için en uygun doğrunun bulunmasında kullanılan en küçük kareler yöntemi gösterilir.
- Bir veri setinden elde edilen bağımlı ve bağımsız değişkenler için bir doğrusal regresyon modeli oluşturulur.
- Oluşturulan doğrusal regresyon modeline göre tahminlerde bulunulur.

**Değerler**

Sabır (Kazanım 5.1., 5.2.)

Özdenetim (Kazanım 5.3.)

**Alan Becerileri**

Tahmin Etme, Kodlama, Programlama Becerisi (Kazanım 5.1., 5.2., 5.3.)

**6. ÜNİTE: VERİ MANİPÜLASYONU****Ünite Açıklaması**

Bu ünite de bir veri setinin kullanılabilir hâle getirilmesi işlemleri üzerinde durulur. Toplulaştırma ve gruplaştırma işlemleri gösterilir. Veri seti üzerinden Label Encoder uygulaması ve One-Hot dönüşümü öğretilir.

**Kazanım ve Açıklamaları****6.1. Veri setleri ile ilgili olan kavramları açıklar.**

Bir veri seti oluşturularak tamlik, teklik, geçerlik, doğruluk ve tutarlılık kavramı açıklanır.

**6.2. Veri setleri üzerinde toplulaştırma ve gruplama işlemleri yapar.**

- Bir veri seti üzerinde `count()`, `first()`, `last()`, `mean()`, `median()`, `min()`, `max()`, `std()`, `var()`, `sum()` komutlarını kullanarak toplulaştırma işlemleri gerçekleştirilir.
- Gruplama işleminin toplulaştırma fonksiyonları ile kullanılması gerektiği vurgulanır.

**6.3. Bir veri setindeki eksik ve aykırı değerleri tespit eder.**

- Eksik ve aykırı değer kavramları açıklanır.
- Aykırı değerleri belirlemenin en kolay yolunun kutu grafiği olduğu bilgisi vurgulanır ve bir liste üzerinden kutu grafiği çizimi gösterilir.
- Aykırı değere ulaşmak için 1.çeyreklik, 3.çeyreklik ve çeyrekler arası açıklık değerleri açıklanır ve `quantile()` komutunun kullanımı gösterilir.
- Aykırı değerlerden kurtulmak için silme ve baskılama yöntemleri gösterilir.
- Veri setinde bulunan eksik değerler `isnull()` komutu kullanılarak gösterilir.
- Eksik verilerin silinmesi için `dropna()` komutu, eksik verinin yerini bir değer ile doldurmak için `fillna()` komutu ve `methods` parametresinin kullanımı gösterilir.

**6.4. One-Hot dönüşümü ile kategorik değişkenleri sayısal değişkenlere dönüştürür.**

- Bir veri setinin doğru yorumlanabilmesi için kategorik değişkenlerin sayısal değişkenlere dönüşümü gösterilir.
- Kategorik değişkenlerin sayısal değişkenlere dönüştürülmesi için `get_dummies()` komutu, `columns` ve `prefix()` parametreleri kullanımı gösterilir.

**6.5. Label Encoder ile kategorik değişkenleri sayısal değişkenlere dönüştürür.**

- Kategorik değişkenlerin sayısal değişkenlere dönüştürülmesi için Label Encoder yönteminin kullanımı gösterilir.
- Bir veri seti üzerinde One-Hot dönüşümü ve Label Encoder yöntemi kullanılarak kategorik değişkenleri sayısal değişkenlere dönüştürülür.
- One-Hot dönüşümü ve Label Encoder yöntemi arasındaki fark açıklanır.

**6.6. Verileri standart hâle getirmek için gerekli olan fonksiyonları kullanır.**

- Verilerin neden standart hâle getirilmesi gerektiği açıklanır.
- Verileri standart hâle getirmek için `StandardScaler()` ve `preprocessing.scale()` komutlarının kullanımı gösterilir.

**6.7. Fonksiyon kullanarak verileri normalleştirir.**

- Verilerin neden normalleştirilmesi gerektiği açıklanır.
- Scikit-learn kütüphanesinin `Preprocessing` modülüne ait `normalize()` fonksiyonunun kullanımı gösterilir.

**Değerler**

Sabır (Kazanım 6.1., 6.2., 6.3., 6.4., 6.5., 6.6., 6.7.)

**Alan Becerileri**

Veri Toplama, İşleme, Dönüştürme Becerisi, Programlama ve Kodlama Becerisi (Kazanım 6.1., 6.2., 6.3.)

Test Etme ve Hata Ayıklama Becerisi (Kazanım 6.4., 6.5., 6.6., 6.7.)



## 7. ÜNİTE: VERİ UYGULAMALARI

### Ünite Açıklaması

Bu ünite de açık kaynak kodlu platformları kullanarak önceki ünitelerde öğrendikleri bilgiler ile bir veri seti üzerinde işlemler yapılması öğretilir.

### Kazanım ve Açıklamaları

#### 7.1. Açık kaynak kodlu platformları kullanır.

- Açık kaynak kodlu platformlar açıklanır.
- Açık kaynak kodlu bir platform üzerinden kullanıcı hesabı oluşturulur.

#### 7.2. Açık kaynak kodlu platformlar üzerinden veri analizi projeleri geliştirir.

- Açık kaynak kodlu platform kullanılarak bir veri seti üzerinden istatistiksel değerlerin bulunması gösterilir.
- Veri indeksleme ve birleştirme, veri setinin görselleştirilmesi, veriler arası ilişkileri inceleme, veri manipülasyonu yöntemlerini kullanarak bir veri setinin analizi gerçekleştirilir.

### Değerler

Sabır, özdenetim (Kazanım 7.1.)

### Alan Becerileri

21. Yüzyıl Becerisi (7.1., 7.2.)

## 8. ÜNİTE: YAPAY ZEKÂ TEMEL KAVRAMLARI VE UYGULAMALARI

### Ünite Açıklaması

Bu ünite de yapay zekâ temel kavramları ve yapay zekâ alt alanları üzerinde durulur. Yapay zekâ ile etik kavramları arasındaki ilişki öğretilir. Model ve tahmin modeli kavramları ile iki veri seti arasındaki korelasyon durumundan bahsedilir. Doğrusal ve çoklu doğrusal regresyon, karar ağacı ve rastgele orman regresyon algoritmaları ile ilgili uygulamalar yapılır. Lojistik regresyon, karar ağacı, rastgele orman, K en yakın komşuluk (KNN), destek vektör makinesi ve Naive Bayes sınıflandırma algoritmaları ile karmaşıklık matrisi konuları üzerinde durulur. Güdümsüz modeller tavsiye sistemleri ve görüntü sıkıştırma algoritmalarından bahsedilir.

### Kazanım ve Açıklamaları

#### 8.1. Yapay zekâyla ilişkili olan kavramları açıklar.

- Makine öğrenmesi, doğal dil işleme ve görüntü işleme kavramları açıklanır.
- Yapay zekâ ve etik arasındaki ilişki açıklanır.
- Yapay zekâ ve veri bilimi arasındaki ilişki açıklanır.
- Ülkemizin savunma sanayisinde kullandığı yapay zekâ uygulamaları üzerinde durulur.

**8.2. İstatistiksel yöntemler ve farklı makine öğrenimi teknikleri kullanarak tahmin modelleri geliştirir.**

- a) Tahmin modeli ve korelasyon kavramı açıklanır.
- b) Bir veri setindeki bağımlı ve bağımsız değişkenler belirlenir.
- c) Bağımlı ve bağımsız değişkenler arasındaki korelasyon değeri bulunur.
- ç) Bağımlı ve bağımsız değişkenler arasında doğrusal bir ilişkinin varlığı hakkında değerlendirme yapılır.
- d) Bağımlı ve bağımsız değişkenler görselleştirilir.
- e) Doğrusal regresyon algoritmasının kullanılabilmesi için Sklearn kütüphanesinin çalışma ortamına aktarılması gösterilir.
- f) Doğrusal regresyon modelinin kurulumu, eğilimi ve y eksenini kestiği noktaları bulmak için `LinearRegression()`, `fit()`, `coef_` ve `intercept_` komutlarının kullanımı gösterilir.
- g) Bir model oluşturulur ve bu model kullanılarak tahminlerde bulunulur.
- ğ) Eğitim verisi ve test verisi kavramları açıklanır.
- h) Veri setini eğitim ve test veri setlerine ayırmak için `train_test_split` fonksiyonunun kullanımı gösterilir.
- ı) Model kullanılarak tahminlerde bulunulur ve gerçek değerler ile karşılaştırılması için gerekli kod satırlarının yazımı gösterilir.
- i) Birden fazla bağımsız değişken arasındaki ilişkiyi belirleyen ve bağımlı değişken için tahminler yapmaya yarayan çoklu doğrusal regresyon uygulamasının kullanımı gösterilir.
- j) Karar ağacı regresyonu açıklanır ve uygulaması gösterilir.
- k) Rastgele orman regresyonu açıklanır ve uygulaması gösterilir.

**8.3. Sınıflandırma modellerini kullanarak belirli sınıflara ait verileri tanımlar.**

- a) Sınıflandırma modeli kavramı açıklanır.
- b) Lojistik regresyon, karar ağacı, rastgele orman, K en yakın komşuluk (KNN), destek vektör makinesi ve Naive Bayes sınıflandırma algoritmaları açıklanır ve uygulaması gösterilir.
- c) Karmaşıklık matrisi kullanarak sınıflandırma probleminin verimliliği gösterilir.

**8.4. Büyük ve karmaşık veri kümeleri içindeki desenleri keşfetmek ve veri setlerindeki bilgileri keşfetmek için güdümsüz modeller kullanır.**

- a) Güdümsüz modeller açıklanır.
- b) Bir veri seti üzerinde kümeleme işlemi gösterilir.
- c) Tavsiye sistemleri ve görüntü sıkıştırma algoritmaları açıklanır.

**Değerler**

Sabır (Kazanım 8.1., 8.2., 8.3.)

Özdenetim (Kazanım 8.4.)

**Alan Becerileri**

Teknik İşlem Becerisi, Algoritma ve Süreçler (Kazanım 8.1., 8.2., 8.3., 8.4.)

## 9. ÜNİTE: YAPAY ZEKÂ TABANLI OYUN GELİŞTİRME

### Ünite Açıklaması

Bu ünite de oyunlarda yapay zekânın nasıl kullanıldığından bahsedilir. Python programlama dilinde oyun oluşturulması ve minimaks fonksiyonunun kullanımı öğretilir. Kural tabanlı, örüntü eşleştirme tabanlı ve yapay zekâ tabanlı sohbet robotlarının Python programlama dili kullanarak nasıl geliştirildiğinden bahsedilir.

### Kazanım ve Açıklamaları

#### 9.1. Oyunlarda yapay zekânın kullanımını açıklar.

- Oyunlarda yapay zekâ kullanımına günümüzden örnekler verilir.
- Oyunlarda yapay zekânın geleceği hakkında fikir yürütülmesi sağlanır.

#### 9.2. Bir yapay zekâ oyun uygulaması geliştirir.

- Tic Tac Toe oyunu anlatılır ve gerekli olan fonksiyonların kullanımı gösterilir.
- Oyuncu olmayan karakterler oluşturulur.
- Recursive tekniği ve minimax algoritması kullanımı gösterilir.
- Derinlik parametresinin eklenmesi gösterilir ve budama kavramı açıklanır.

#### 9.3. Yapay zekâ tabanlı Chatbot uygulamaları geliştirir.

- Sohbet robotlarının nasıl çalıştığı açıklanır.
- Kural tabanlı sohbet robotu geliştirilir.
- Örüntü eşleştirme tabanlı sohbet robotu geliştirilir.
- Kullanıcının duygu durumunu analiz eden bir uygulama gösterilir.
- Sohbet robotu oluşturmak için Chatterbot kütüphanesinin kullanımı gösterilir.
- Sohbet robotunu eğitmek için ListTrainer modülünün kullanımı gösterilir.
- Atatürk'ün hayatı ile ilgili bir dizi soruya cevap verebilen sohbet robotu uygulaması gerçekleştirilir.

### Değerler

Sabır (Kazanım 9.2., 9.3.)

Özdenetim (Kazanım 9.3.)

### Alan Becerileri

Kodlama ve Programlama Becerisi, Sistem Kurma (İnşaa Etme) Becerisi, 21. Yüzyıl Becerileri (Kazanım 9.1., 9.2., 9.3.)