

Programlama Dilleri

Grup/dil dağıtım listesinde grubunuza atanan programlama diliyle aşağıdaki başlıklandırılan üç farklı çalışmayı sunum haftasına kadar tamamlayacaksınız.

Kurallar

- Gruplama sabittir. Değiştirilmesini talep edemezsiniz.
- Sunum haftası sabittir. Tarihi öne veya geri alamazsınız. Sıranız geldiğinde size atanan konuyu sunmak zorundasınız.
- Grup içinde başlık seçimi de (dil bilgisi, problem ve test) sabittir. Değiştirilmesi talebinde bulunamazsınız.
- Başlık paylaşılması yapılmakla birlikte grup üyelerinin tümü başlıkların hepsinden sorumludur. Grupta sunum sırasında tek başınıza kalsanız bile tüm başlıkları sunmak zorundasınız.
- Her grup için 45 dk ayrılmıştır. Bu süre üç başlık arasında eşit şekilde (15 dk) paylaşılacaktır. Sunumunuzu size ayrılan bu süre içinde tamamlamanız gerekiyor.
- Tüm kodlar Github'ta bir depoda bulunmalı. Problem ve test grupları tek bir depoyu, tercihen biri diğerinden fork ederek, kullanabilir.
- Sunum öncesinde projeksiyon cihazının ve ihtiyaç varsa ağ bağlantısının hazır hale getirilmesinden sorumlusunuz. Sunum öncesinde erken gelerek hazırlığınızı yapın. İşinizi son dakikaya bırakmayın ve bir başkasından yardım istemeyin. **Zamanında hazırlık yapmadığından sunumu aksayan grupların dönem içi puanları en az %25 oranında kırılacaktır.**
- Size atanan dil ve konuyla ilgili yüz yüze veya eposta yoluyla sürekli soru üretmeyin (ör. “Hocam şöyle mi yapayım? Böyle mi yapayım?”) Her detayı dokümanite edemiyoruz. Belirsizlik olan konularda dürüst şekilde inisiyatif kullanın. Sunumlar ilerledikçe varsa belirsizlikler ders içinde yapılan açıklamalarla giderilecektir.

- Haftalar ilerledikçe sunum sırasında gördüğüm problemleri dikkate alarak yeni kurallar koyabilirim. Bu nedenle her derse katılmak ve bu açıklamaları not ederek kendi sunumunuza uygulamakla sorumlusunuz.

Başlıklar

Dil bilgisi

Programlama dilini **basit örneklerle** tanıtan kısa bir sunum hazırlayın. Şu bilgiler mutlaka olmalı.

1. Dil hangi tarihte ve kimler tarafından geliştirilmiş?
2. Temel söz dizimi (basit bir “merhaba dünya” programı üzerinde)
3. Dilin doğruluk/yanlışlık (truthy/falsy) kuralları
4. Değişken kapsam (scope) kuralları neler? Yerel ve genel değişkenler.
5. Fonksiyon/metod nasıl tanımlanıyor? Parametre aktarımı ve dönüş değerleri.
6. Nesne yönelimli programlama özellikleri? Sınıf nasıl tanımlanıyor? Nesne nitelikleri, nesne metodları ve sınıf metodları nasıl tanımlanıyor?
7. Çok kısaca dille birlikte gelen standart kitaplıkta dizgi (string), dizi (array) ve ilişkili dizi (hash) işlemlerini örneklendirin.

Dikkat! Tüm bilgiler kısa örnekler üzerinden somut şekilde anlatılmalı. Teorik açıklama yok.

Problem

4 yıllık hayali bir ilköğretim okulu için dosya tabanlı çok basit bir veritabanı uygulaması yazın.

- Toplamda 4 farklı devre bulunuyor: birinci, ikinci, üçüncü ve dördüncü (son sınıflar).
- Bir öğrenci ad, soyad, devre (1, 2, 3 ve 4) ve cinsiyet (E ve K) bilgileriyle tanımlanıyor.
- Başlangıçta tüm devreleri ve farklı cinsiyetten öğrencileri isim çakışması olmadan rastgele şekilde içeren en az 20 kayıtlı bir CSV dosya oluşturun. Bu dosyayı ilk kayıt için kullananın. Örnek içerik (4 kayıt):

```
ad,soyad,cinsiyet,devre
Ahmet,Fuat,E,3
Ayşe,Yılmaz,K,1
Fatma,Demir,K,4
```

Mehmet,Çelik,E,2

Bu bilgilere göre okul isimli bir program yazın.

- Program başlangıçta çalıştığı dizinde `okul.csv` isimli bir dosyayı yüklüyor.
- Yükleme sırasında isim çakışması varsa hata verip çıkıyor.
- Argüman verilmediğinde okuduğu kayıtları sırasıyla devre, ad ve soyad sırasıyla görüntülüyor.

```
$ okul
1 Ayşe Yılmaz K
2 Mehmet Çelik E
3 Ahmet Fuat E
4 Fatma Demir K
```

- Argüman olarak sayı girildiğinde sayıyı devre numarası olarak yorumluyor ve sadece o devredeki öğrencileri ad soyad sırasıyla görüntülüyor

```
$ okul 3
Ahmet Fuat E
```

```
$ okul 5
Hata: devre numarası 1-4 aralığında olmalı
```

- Argüman olarak `e` veya `k` girildiğinde (küçük büyük harf duyarlı) sadece ilgili cinsiyetteki öğrencileri sırasıyla devre, ad ve soyad sırasıyla görüntülüyor.

```
$ okul k
1 Ayşe Yılmaz
4 Fatma Demir
```

- Bunun dışındaki tüm argümanlarda hata veriyor

```
$ okul erkek
Hata: tanınmayan argüman: erkek
```

```
$ okul 2 k
Hata: sadece bir argüman girebilirsiniz
```

Test

Yazılan program için ilgili programlama dilindeki en yaygın ve standart test kütüphanesini kullanarak testler yazın. Yazılacak testler yukarıda örneklenen tüm durumları test etmeli. Eksik test kabul edilmeyecektir.

“Yaygın ve standart” test kütüphanesi dilden dile değişiklik gösterir. Örneğin Ruby’de “mini-test” veya “rspec” kullanılır. Go programlamada dilinde test kütüphanesi

zaten hazır geldiğinde ayrıca bir seçim yapmaya gerek yoktur. Bash için “Bats” kullanılabilir vs.