

## PROGRAMLAMA ÖDEV GRUBU ÖDEV:3

SİNEM GÜNDÜZALP 20200805004

### SORU 1)

Yazılımın doğru çalışmasını engelleyen hataların bütünü yazılımsal hatadır. Bu hatalar mantık, yazım hatası, kural dışı eylem, arıza ve daha birçok sebepten oluşabilir.

1)Syntax error (söz dizilimi hataları), geliştirilen programda, kullanılan programlama dilinin kurallarına aykırı ifadelerin yer almasından dolayı ortaya çıkar.

2)Runtime error, (yani çalışma zamanı hataları) olmayan bir dosyanın açılması sırasında veya olmayan bir dosyanın üzerine veri yazmaya çalışıldığı zaman ya da işletim sisteminin yapamayacağı şeylerin istenmesi durumunda ortaya çıkan hata türüdür.

3)Logic Error, ( mantık hataları) programın yanlış çalışmasına neden olan, istenmeyen çıktılar üreten mesela  $a=50$   $b=60$  için denklem  $a + b / 2$ ; şeklinde yazılır ise çıktı 80 olur ve mantık hatası yapılmış olunur çünkü  $50 + 60$  işleminin sonucu 110'dur. 110'u 2'ye böler isek sonuç 55 olmalıdır. Bu bir mantık hatası örneğidir

4)Compilation Error (Derleme veya derleme zamanı hatası) derleyici kodu alt seviye dile nasıl dönüştüreceğini bilmediğinde meydana gelir. Derleme hataları ile söz dizilimi hataları karıştırılmamalıdır. Derleme hataları programın derlenmesi sırasında meydana gelir, dolayısıyla program derlenene kadar hata ortaya çıkmaz. Program derlendikten sonra hata olup olmadığı konsol ya da çıktı sayesinde öğrenilebilir.

5)Kaynak Hataları ise bir program, üzerinde çalıştığı bilgisayarı sahip olduğundan daha fazla kaynak (işlemci gücü, RAM, disk alanı vb) ayırmaya zorlarsa ortaya çıkar. Bu durum programın tıkanmasına veya tüm sistemin çökmesine yol açabilir.

6)Arayüz Hataları programın nasıl kullanılmasını amaçladığınız ile gerçekte nasıl kullanıldığı arasında bir bağlantı olmadığında ortaya çıkan hata türüdür.

7)Aritmetik hatalar, Logic Error yani mantık hatalarının bir diğer türüdür. Örneğin bir çarpma denklemi gerçekleştirirken yanlış sayıya çarpılan değişkendeki veri yüzünden tüm programdaki çıktı hatalı olabilir.

KOD yazmada hatayı azaltmak için: Kodlama öncesinde problemi iyi anlayıp ve tasarımına dikkat etmeliyiz, düzenli olarak kodu test edebiliriz parça parça halinde tüme ulaşmadan kontrol etmek bize kolaylık sağlar. Debuggerları kullanıp hataları test edip ayıklayabiliriz, sürüm kontrol sistemi kullanabiliriz ve de en önemlisi kodu açıklayıcı biçimde yazmalıyız ki başkaları kodu daha rahat anlasın ve hataları daha kolay düzeltebilsin.

**SORU2) ---**

### ***SORU3)***

```
import javax.swing.JOptionPane;
import java.util.Random;

public class RandomNumberGame {

    public static void main(String[] args) {

        Random random = new Random();

        int min = 1;

        int max = 100;

        int randomNumber = random.nextInt(max - min + 1) + min;

        int attempts = 0

        JOptionPane.showMessageDialog(null, "1-100 arasında rastgele bir sayı tuttum. Tahminin nedir.");

        while (true) {

            try {

                String userInput = JOptionPane.showInputDialog("Tahminin: ");

                if (userInput == null) {

                    // Kullanıcı iptal tuşuna bastı

                    JOptionPane.showMessageDialog(null, "Oyun sonlandırıldı.");

                    break;

                }

                int guess = Integer.parseInt(userInput);

                attempts++;

                if (guess < min || guess > max) {

                    JOptionPane.showMessageDialog(null, "Lütfen 1 ile 100 arasında bir sayı girin.");

                } else if (guess > randomNumber) {

                    JOptionPane.showMessageDialog(null, "Daha düşük bir sayı tuttum.");

                } else if (guess < randomNumber) {

                    JOptionPane.showMessageDialog(null, "Daha yüksek bir sayı tuttum.");

                } else {

                    JOptionPane.showMessageDialog(null, "İşte bildin! " + attempts + " denemede doğru tahmin ettin.");

                    break;

                }

            } catch (NumberFormatException e) {

                JOptionPane.showMessageDialog(null, "Geçerli bir sayı girin.");

            }

        }

    }

}
```

#### SORU4)

```
public class ABCPrimes {  
    public static void main(String[] args) {  
        for (int abc = 100; abc <= 999; abc++) {  
            int cba = reverseNumber(abc); // sayıları ters çevirmesi için reverse kullandık  
            // isPrime ise asal sayı kontrolü yapar  
            if (cba > abc && isPrime(abc) && isPrime(cba) && isPrime(abc / 100) && isPrime((abc % 100) / 10) &&  
isPrime(abc % 10)) {  
                System.out.println(abc);  
            }  
        }  
    }  
}  
  
// Sayıyı ters çevirelim  
public static int reverseNumber(int num) {  
    int reversed = 0;  
    while (num != 0) {  
        int digit = num % 10;  
        reversed = reversed * 10 + digit;  
        num /= 10;  
    }  
    return reversed;  
}  
  
// Asal sayı kontrol edelim  
public static boolean isPrime(int num) {  
    if (num <= 1) {  
        return false;  
    }  
    for (int i = 2; i * i <= num; i++) {  
        if (num % i == 0) {  
            return false;  
        }  
    }  
    return true;  
}  
}
```

### SORU5)

```
public class PrimeChecker {

    public static void main(String[] args) {

        int num = 113; // Test edilecek sayı

        boolean isPrimeXResult = isPrimeX(num);

        System.out.println("isPrimeX(" + num + ") = " + isPrimeXResult);

    }

    // Verilen sayının asal olup olmadığını yine bu methodla kontrol ederiz

    public static boolean isPrime(int num) {

        if (num <= 1) {

            return false;

        }

        for (int i = 2; i * i <= num; i++) {

            if (num % i == 0) {

                return false;

            }

        }

        return true;

    }

    // isPrimeX metodu için

    public static boolean isPrimeX(int num) {

        while (num >= 10) {

            if (!isPrime(num)) {

                return false;

            }

            int sum = 0;

            while (num > 0) {

                sum += num % 10;

                num /= 10;

            }

            num = sum;

        }

        return isPrime(num);

    }

}
```