Döngüler - 2

- Genel bir while döngüsü şu yapıya sahiptir:

```
while (koşul) {

    // yürütülecek kod

    // koşulu güncelle
}
```

- Bir 'sentinel' döngüsü, girilen özel bir değere kadar devam eden özel bir while döngüsüdür. Örneğin, aşağıdaki kod, kullanıcı -1'i girilene kadar bir sayı girmeye devam edecektir.

```
Scanner scanner = new Scanner(System.in);
int number = 0;
while (number != -1) {
    System.out.print("Bir sayı girin (-1 çıkış için): ");
    number = scanner.nextInt();
}
```

- do while döngüleri, koşulun döngünün başında değil de sonunda kontrol edildiği while döngülerine benzer. Bu, döngünün her zaman en az bir kez çalışacağı anlamına gelir.

```
do {
    // yürütülecek kod
} while (koşul);
```

- for döngüleri, kaç kez döngünün çalışmasını istediğinizi bildiğinizde kullanılan özel bir döngü türüdür. For döngüsünün genel yapısı şöyledir:

```
for (başlatma; koşul; güncelleme) {
    // yürütülecek kod
}
```

- `break`, bir döngüden (veya herhangi bir diğer kod bloğundan) çıkmak için kullanılabilen bir anahtar kelimedir. Örneğin, aşağıdaki kod, kullanıcı -1'i girene kadar bir sayı girmeye devam edecek, ardından döngüden çıkacaktır.

```
import java.util.Scanner;

public class punchcard {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.print("Çıkış için bir sayı girin (-1): ");
            int number = scanner.nextInt();
            if (number == -1) {
                break;
            }
        }
        System.out.println("Hoşçakal!");
    }
}
```

- `continue`, bir döngünün mevcut iterasyonunun geri kalanını atlamak için kullanılabilen bir anahtar kelimedir. Örneğin, aşağıdaki kod 1'den 10'a kadar sayacı sayacak ancak 5 sayısını atlayacaktır.

```java
public class punchcard {
    public static void main(String[] args) {
        int count = 0;
        while (count < 10) {
            count++;
            if (count == 5) {
                continue;
            }
            System.out.println(count);
        }
    }
}
```

1. While döngüsünün yalnızca 10 kez çalışmasını sağlamak için boşluğu doldurun.

```
public class punchcard {
    public static void main(String[] args) {
        int i = 1;
        while (______) {
            System.out.println("Döngüye " + i + " kez girildi");
            i++;
        }
    }
}
```

a) `i < 9`
b) `i <= 9`
c) `i < 10`
d) `i <= 10`
e) `i <= 11`

cevap: d - `i <= 10`

---

2. Aşağıdaki döngü kaç kez çalışacaktır?

```
public class punchcard {
    public static void main(String[] args) {
        boolean flag = false;
        while (flag) {
            System.out.println("Döngü içinde");
        }
        System.out.println("Döngüden sonra");
    }
}
```

a) 0
b) 1
c) Sonsuz
d) Derleme hatası
e) Çalışma zamanı hatası

cevap: a - 0

---

3. Aşağıdaki döngü kaç kez çalışacaktır?

```
public class punchcard {
    public static void main(String[] args) {
        do {
            System.out.println("Döngü içinde");
        } while (false);
        System.out.println("Döngüden sonra");
    }
}
```

a) 0
b) 1
c) Sonsuz
d) Derleme hatası
e) Çalışma zamanı hatası

cevap: b - 1

---

4. For döngüsünün yalnızca 5 kez çalışmasını sağlamak için boşluğu doldurun.

```
public class punchcard {
    public static void main(String[] args) {
        int i = 0;
        do {
            System.out.println("Döngü içinde");
            ---------- {
                break;
            }
            i++;
        } while (true);
        System.out.println("Döngüden sonra");
    }
}
```

a) `if (i <= 4)`
b) `if (i == 4)`
c) `while (i < 4)`
d) `while (i <= 4)`
e) `else if (i <= 4)`

cevap: b - `if (i == 4)`

---

5. Aşağıdaki döngü kaç kez çalışacaktır?

```
public class punchcard {
    public static void main(String[] args) {
        for (int i = 0; i < 10; i=i+2) {
            System.out.println("Döngü içinde");
        }
        System.out.println("Döngüden sonra");
    }
}
```

a) 0
b) 1
c) 5
d) 10
e) Sonsuz

cevap: c - 5

---

6. Aşağıdaki döngü kaç kez çalışacaktır?

```
public class punchcard {
    public static void main(String[] args) {
        boolean flag = true;
        for (int i = 0; flag; i++) {
            System.out.println(i + ". kez döngü iç

inde");
            if (i*i == 64) {
                flag = false;
            }
        }
    }
}
```

a) 0
b) 1

c) 4
d) 8
e) 9

---

7. Muhasebe yazılımı için bir menü sistemine sahip olmak istiyorum. Kullanıcı 0'ı girene kadar menü seçeneği sormaya devam etmek istiyorum. Aşağıdaki kodun ne gibi bir sorunu var?

```
import java.util.Scanner;

public class punchcard {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        do {
            int number = scanner.nextInt();
            switch(number) {
                case 1:
                    System.out.println("Dil seçimi");
                    break;
                case 2:
                    System.out.println("Müşteri destek");
                    break;
                case 3:
                    System.out.println("Bakiyeyi kontrol et");
                    break;
                case 4:
                    System.out.println("Kredi bakiyesini kontrol et");
                    break;
            }
        } while(number != 0);
        System.out.println("Çıkış");
    }
}
```

a) `scanner` nesnesi doğru şekilde başlatılmamıştır.
b) switch ifadesinde varsayılan durum yok.
c) `number` değişkeni do-while döngüsünden önce tanımlanmamıştır.
d) Döngü koşulu, kullanıcının 0 girdiğinde çıkmasına izin vermiyor.
e) Kod istenildiği gibi çalışmalıdır. Hiçbir sorun yok.

cevap: c - `number` değişkeni do-while döngüsünden önce tanımlanmamıştır.

---

8. Aşağıdaki kodun çalıştırılması durumunda çıktısı ne olur?

```
public class punchcard  {
    public static void main(String[] args) {
        String message = "Merhaba, Dünya!";
        for (int i = 1; i < message.length(); i = i*2) {
            char currentChar = message.charAt(i);
            if (Character.isLetter(currentChar)) {
                System.out.print(Character.toUpperCase(currentChar));
            } else {
                System.out.print(currentChar);
            }
        }
    }
}
```

a) Merhaba, Dünya!
b) mERHABA, dÜNYA!

c) M, D!
d) MERHABADÜNYA
e) ERAB

cevap: e - ERAB

---

9. Aşağıdaki kod ne yapar?

```
public class punchcard  {
    public static void main(String[] args) {
        String input = "Java Programlama";
        String reversed = "";

        for (int i = input.length() - 1; i >= 0; i--) {
            reversed += input.charAt(i);
        }

        System.out.println(reversed);
    }
}
```

a) Orijinal dizesini değiştirmeden yazdırır.
b) Yanlış döngü uygulaması nedeniyle bir hata üretir.
c) Dizideki karakterleri ters çevirir ve sonucu yazdırır.
d) Dizeden tüm ünlüleri kaldırır.
e) Dizedeki karakterleri tüm büyük harflerle yazdırır.

cevap: c - Dizideki karakterleri ters çevirir ve sonucu yazdırır.

---

10. Aşağıdaki kod çalıştırıldığında çıktısı ne olur?

```
public class StringModification {
    public static void main(String[] args) {
        String text = "Java'ya Hoş Geldiniz";
        String modifiedText = "";

        for (int i = 0; i < text.length(); i++) {
            char currentChar = text.charAt(i);
            if (currentChar != ' ') {
                modifiedText += Character.toUpperCase(currentChar);
            } else {
                modifiedText += "\b";
            }
        }

        System.out.println(modifiedText);
    }
}
```

a) JAVA'YA HOŞ GELDINIZ
b) Java'YaHoşGeldiniz
c) Java'ya Hoş Geldiniz
d) javaYOHoşGeldiniz
e) JAVA YA HOŞ GELDINIZ

cevap: e - JAVA YA HOŞ GELDINIZ

---

11. Aşağıdaki kod çalıştırıldığında çıktısı ne olur?

```java
public class punchcard {
    public static void main(String[] args) {
        String sentence = "Java eğlencelidir. java eğlencelidir. JavaEğlencelidir. Jav eğlencelidir.";
        int count = 0;

        for (int i = 0; i < sentence.length() - 3; i++) {
            // if (sentence.substring(i, i + 4).equals("Java")) {
            //  count++;
            // }
            int index = sentence.indexOf("Java", i);
            if (index != -1) {
                count++;
                i = index;
            }
        }

        System.out.println("The word 'Java' appears " + count + " times.");
    }
}
```

a) 1
b) 2
c) 3
d) 4
e) Hata

cevap: b - 2

---

12. Aşağıdaki döngünün sonlandırma koşulu nedir?

```java
public class punchcard {
    public static void main(String[] args) {
        String string = "ZERO-ONEE-ZERO-ZERO-ONEE-ZERO-ZERO-ZERO-ZERO-ZERO-ONEE-ONEE-ZERO-ONEE-ZERO-ZERO-ZERO-ON
        int index = 0;
        while (________________________) {
            int nextHyphen = string.indexOf("-", index);
            if (nextHyphen == -1) {
                nextHyphen = string.length();
            }
            String number = string.substring(index, nextHyphen);
            switch (number) {
                case "ZERO":
                    System.out.print("0");
                    break;
                case "ONEE":
                    System.out.print("1");
                    break;
            }
            index = nextHyphen + 1;
        }
    }
}
```

a) `index < string.length()`
b) `index < string.indexOf("-")`
c) `index < string.indexOf("-", i)`
d) `index < string.length() - 4`
e) `index < string.indexOf("-ONEE\n")`

cevap: d - `i < string.length() - 4`

NOT: Bu problem için string.split() kullanmak daha iyi bir çözüm olacaktır. Ancak henüz bunu kapsamadık, bu yüzden yukarıdaki çözümü kullanacağız.

---

13. Önceki kodun çıktısı hala anlamsız. Şifre çözme işlemine devam etmek için, her 8 bitin arasına bir boşluk eklememiz gerekiyor.

Aşağıdaki boşluğa ne gelmelidir böylece kod parçası her 8 bitin ardından bir boşluk ekleyebilsin?

```
public class PunchCard {
    public static void main(String[] args) {
        String string = "01001000001101000100001101001011001100110101001000110101";
        int i;
        for (i = 0; _____________________________) {
            System.out.print(string.substring(i, i + 8) + " ");
        }
    }
}
```

a) `i + 8 <= string.length(); i += 8`
b) `i < string.length(); i ++`
c) `i + 8 <= string.length(); i ++`
d) `i > string.length(); i += 8`
e) `i == string.length(); i ++`

cevap: a - `i + 8 <= string.length(); i += 8`

İkili kodu ondalığa çevirmek için aşağıdaki kod parçasını kullanmamız gerekiyor:

Aşağıdaki kodda kullanılan bazı yöntemleri kapsamadığımızdan, bunları açıklama olmaksızın kullanacağız.

```
public class BinaryToDecimal {
    public static void main(String[] args) {
        String binaryString = "01001000 00110100 01000011 01001011 00110011 01010010 00110101";

        // İkili dizgiyi ayrı ikili sayılara böler
        String[] binaryNumbers = binaryString.split(" ");

        // Her ikili dizgiyi ondalık sayıya dönüştür ve yazdır
        for (String binary : binaryNumbers) {
            int decimal = Integer.parseInt(binary, 2);
            System.out.print(decimal + " ");
        }
    }
}
```

çıktı:

```
72 52 67 75 51 82 53
```

---

14. Şimdi ondalık sayılarımız var, bunları ASCII karakterlere dönüştürmemiz gerekiyor. Aşağıdaki kod parçasında ne yanlış?

```
public class PunchCard {
    public static void main(String[] args) {
        String input = "72 52 67 75 51 82 53";

        for (i = 0; i < input.length(); i += 3) {
            System.out.print((char) Integer.parseInt(input.substring(i, i + 2)));
        }
    }
}
```

çıktı:

```
H4CK3R5
```

a) for döngüsünün güncelleme ifadesi `i += 2` olmalıdır.
b) Substring yöntemi `i + 3` kullanmalıdır, `i + 2` yerine (`input.substring(i, i + 2)`)
c) İterasyon değişkeni `i` tanımlanmalıdır.
d) for döngüsünün koşulu `i < input.length() - 2` olmalıdır, son karakterde durmak için.
e) Kodun hiçbir yanlışı yoktur.

cevap: c - İterasyon değişkeni `i` tanımlanmalıdır.

---

15. Şifre çözümünde oldukça ilerledik. Ancak hala bir sorunumuz var. Kimse leet dilini bilmiyor, ancak karakterlere sahibiz. Leet karakterlerini normal karakterlere dönüştürmemiz gerekiyor.

sözlük:

```
4 = A
8 = B
3 = E
6 = G
1 = I
0 = O
2 = Z
```

Aşağıdaki kod parçası çalıştırıldığında çıktısı ne olacaktır?

```
public class PunchCard {
    public static void main(String[] args) {
        String input = "H4CK3R5";

        for (int i = 0; i < input.length(); i++) {
            char currentChar = input.charAt(i);
            switch (currentChar) {
                case '4':


    System.out.print("A");
                    break;
                case '8':
                    System.out.print("B");
                    break;
                case '3':
                    System.out.print("E");
                    break;
                case '6':
                    System.out.print("G");
                    break;
                case '1':
                    System.out.print("I");
                    break;
                case '0':
                    System.out.print("O");
                    break;
                case '2':
                    System.out.print("Z");
                    break;
                default:
                    System.out.print(currentChar);
            }
        }
```

```
    }
}
```

a) H4CK3R5
b) H4CEKR5
c) HAGKEBZS
d) HZCK3R5
e) HACKER5

cevap: e - HACKER5