Ödev 3 - Rapor

17253509
Bedrettin Bora Tanrıkulu
Bilgisayar Mühendisliği - 1.Sınıf

Tortoise and Hare

- Proje 4 class'tan oluşmaktadır.
  Bunlar;
  1) RaceTest
  2) Tortoise
  3) Hare
  4) Race

RaceTest Class'ı

- Projenin test edilmesi için oluşturulmuş bir class'tır. Tortoise ve Hare sayıları kullanıcıdan alındıktan ve gerekli kontroller yapılır. Ardından **race.startRace();** ile yarış başlatılır.

  > Bu class'ta bunun haricinde başka bir işlem yapılmamaktadır.

```java
public class RaceTest {

    public static void main(String[] args) {
        // objects
        Scanner scan = new Scanner(System.in);
        Race race = new Race();

        // questions
        System.out.print("How many tortoises? ");
        int tortoiseNumber = scan.nextInt();
        System.out.print("How many hares? ");
        int haresNumber = scan.nextInt();

        // control and settings
        if(tortoiseNumber < 0 || haresNumber < 0)
            System.out.println("(!) Do not use negative values!");

        else if(tortoiseNumber == 0 && haresNumber == 0)
            System.out.println("(!) There are no runners!");

        else {
            race.setTortoises(tortoiseNumber); // sends the number of tortoise to the race object.
            race.setHares(haresNumber); // sends the number of hares to the race object.

            race.startRace(); // starts the race!.
        }
    }
}
```

Tortoise ve Hare Class'ları

- Tortoise ve Hare'in özelliklerini içeren class'lardır. Yapı olarak birbirinin aynısı sayılabilir. 1-10 arasında, **SecureRandom** ile, rastgele bir sayı üretilip Tortoise ve Hare'in özelliklerine uygun hareket tipi tespit edilip **return** ile geri gönderilir.

  > Yarış alanının dışına çıkılmasını engellemek amaçlı; 0'dan küçük değerleri 0'a, 69'dan büyük değerleri 69'a eşitler.

```java
public class Tortoise {
    private int move;
    private int chance;

    // setters
    public void setChance() {
        SecureRandom srandom = new SecureRandom();
        this.chance = srandom.nextInt(10) + 1; // generates a random number
    }
    public void setMove() { // changes the "move" with the move type information.
        if(this.chance>0 && this.chance<=5)
            this.move += 3;
        else if(this.chance>5 && this.chance<=7)
            this.move += -6;
        else if(this.chance>7 && this.chance<=10)
            this.move += 1;
    }

    // getter
    public int getMove() {
        if(this.move < 0) // shows 0 if it is less than zero.
            this.move = 0;

        if(this.move >= 69) // shows 69 if it is greater than 69.
            this.move = 69;

        return this.move;
    }
}
```

Race Class'ı

- Yarışın gerçekleştiği class'tır. İçersinde 4 method bulunur. Bunlar;
  1) startRace
  2) isItWinner
  3) showWinners
  4) display

startRace Method'u

- Yarışın gerçekleştiği kısımdır. Diğer method'lar çağırılıp burada kullanılır. Gerekli array oluşturmaları yapıldıktan sonra; kazanan biri olmadıkça döngüde kalan bile **while()** içersinde bulunan **for(;;)**'lar ile yarış işlemi gerçekleştirilir ve aynı zamanda adım adım yarış ekrana basılır.

  Eğer herhangi bir kazanan olursa o anki tur bittikten sonra döngüden çıkılır ve kazananlar ekrana basılır.

```java
        // objects
        for (int i=0; i < hare.length; i++) // builds the hare objects that defined as an array.
            hare[i] = new Hare();
        for(int i=0; i < tortoise.length; i++) // builds the tortoise objects that defined as an array.
            tortoise[i] = new Tortoise();

        while(winnersNumber<=0) {
            System.out.printf("Round %d\n", (++counter));

            // race and display loops
            for(int i=0; i < tortoise.length; i++) {
                tortoise[i].setChance();
                tortoise[i].setMove();
                display(tortoise[i].getMove(), 'T'); // shows the current tortoise runner position with "T".
            }
            for(int i=0; i < hare.length; i++) {
                hare[i].setChance();
                hare[i].setMove();
                display(hare[i].getMove(), 'H'); // shows the current hare runner position with "H".
            } System.out.println("\n");

            // control loops for the winner
            for (int i=0; i < tortoise.length; i++) {
                if(isItWinner(tortoise[i].getMove())) {
                    tortoiseWinners.add(i); // sends the current tortoise to the arraylist if the is a winner.
                }
            }
            for (int i=0; i < hare.length; i++) {
                if(isItWinner(hare[i].getMove())) {
                    hareWinners.add(i); // sends the current hare to the arraylist if it is a winner.
                }
            }
        }
```

isItWinner Method'u

- Method'a yollanılan yarışcının kazanan olup olmadığını söyler.

> Aynı zamanda **winnersNumber** değerini artırır. Bu sayede **startRace**'de bulunan **while()** döngüsü kırılmış olur.

```java
    public boolean isItWinner(int position){
        if(position == (RACE_LENGTH-1)) { // checks winning situations.
            this.winnersNumber++;
            return true;
        }
        return false;
    }
```

showWinners Method'u

- Yarış bittikten sonra kazanan yarışçıları ekrana basar.

> Birden fazla kazanan olup olmadığını kontrol edilir, **beraberlik durumu** tespit edilirse; ona göre bir işlem uygulanır.

```java
    public void showWinners(ArrayList<Integer> tortoiseWinners, ArrayList<Integer> hareWinners) {
        if(winnersNumber > 1) { // If there are more than one winner.
            System.out.print("It's a tie between");
            for (int i=0; i<tortoiseWinners.size(); i++) {
                System.out.print(" Tortoise[" + ((tortoiseWinners.get(i))+1) + (i==(tortoiseWinners.size()-1) ? ("] at Round "+(counter)) : "],"));
            }
            for (int i=0; i<hareWinners.size(); i++) {
                System.out.print(" Hare[" + ((hareWinners.get(i))+1) + (i==(hareWinners.size()-1) ? ("] at Round "+(counter)) : "],"));
            }
            System.out.println();
        }

        else { // If there is only one winner.
            for (int i=0; i<tortoiseWinners.size(); i++) {
                System.out.println("Tortoise[" + ((tortoiseWinners.get(i))+1) + "] won" + (" at Round "+(counter)) + "! YAY!!");
            }
            for (int i=0; i<hareWinners.size(); i++) {
                System.out.println("Hare[" + ((hareWinners.get(i))+1) + "] won" + (" at Round "+(counter)) + "! YAY!!");
            }
        }
    }
```

display Method'u

- Method'a yollanılan yarışçının 70'lik bir yarış alanındaki konumunu ekrana basar.

> **startRace** method'unda bulunan **while()** döngüsünde her adımda bu method kullanılarak anlık olarak yarışçı konumları ekrana basılır.

```java
    public void display(int location, char c) {
        for(int i=0; i<RACE_LENGTH; i++) {
            if(location == i)
                System.out.printf("%s", c); // shows "H" or "T" if the "i" is current position of runner.
            else
                System.out.printf("_"); // shows "_" if the "i" is not current position of runner.
        }
        System.out.println();
    }
```