

Nama : Furqan Ramadhan

NPM : 2108107010013

Tugas 3 Pemrograman Berorientasi Objek

Deskripsi Tugas

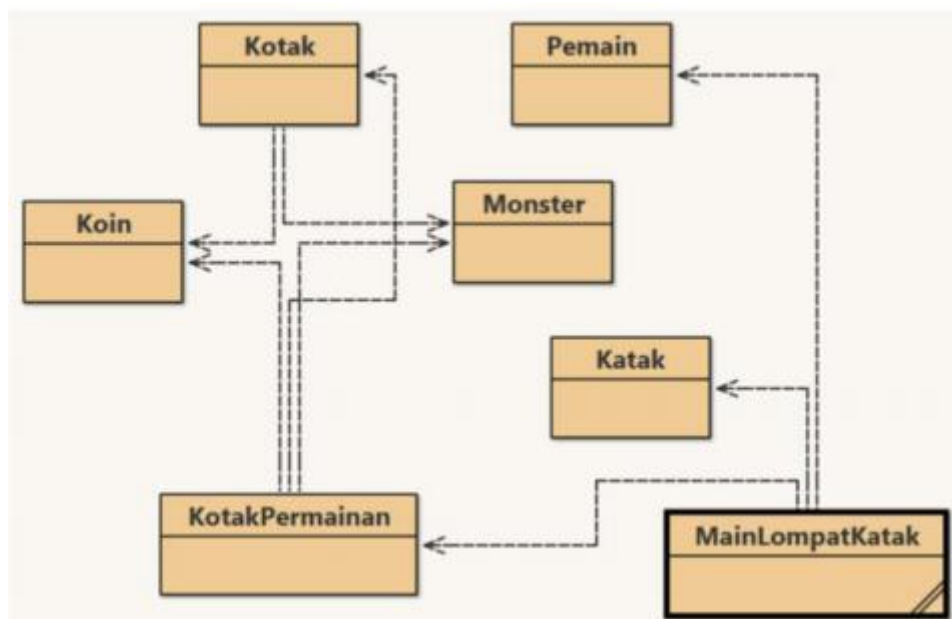
Pada tugas ini anda diharapkan untuk mengubah permainan “Lompat hai katak, lompat!” yang telah anda selesaikan pada Tugas 2. Perubahan tersebut dapat berupa pemanfaatan konsep:

- inheritance,
- interface,
- abstract class,
- exception,
- collection,
- annotation,
- generic
- dan lainnya.

Anda tidak harus menggunakan semua konsep tersebut namun minimal ada 2 konsep yang digunakan untuk merubah Tugas 2 sebelumnya.

Jawab

Konsep awal permainan “Lompat hai katak, lompat!” yang telah dikerjakan pada tugas 2 sebelumnya.



Pada tugas 2 lalu, sudah digunakan beberapa konsep OOP yaitu:

- Exception, pada class MainLompatKatak

```
try {  
    if (decision != 'a' && decision != 'd' && decision != 'A' && decision != 'D') {  
        System.out.println("Insert according to instructions.");  
        continue;  
    }  
} catch (Exception e) {  
    System.out.println("Insert only a or d!");  
}
```

- Encapsulation, pada hampir di semua class kecuali MainLompatKatak

```
public Pemain(String nama) {  
    this.nama = nama;  
    // nilai awal  
    nilai = 0;  
}
```

- Data Collection, pada class KotakPermainan

```
private void generateAcak() {  
    ArrayList<Integer> list = new ArrayList<Integer>();  
    acakKoin = new int[jumlahKoin];  
    acakMonster = new int[jumlahKoin];  
    // nilai acak ditentukan sebanyak jumlah kotak  
    for (i = 0; i < jumKotak; i++) {  
        list.add(i);  
    }  
    // Collection untuk mengacak nilai kepada koin  
    Collections.shuffle(list);  
    // generate nilai acak untuk koin  
    for (i = 0; i < jumlahKoin; i++) {  
        acakKoin[i] = list.get(i); // koin  
    }  
    // generate posisi monster dimana posisi monster != posisi koin  
    for (i = jumlahKoin; i < jumlahMonster + jumlahKoin; i++) {  
        acakMonster[i - jumlahKoin] = list.get(i);  
    }  
}
```

Pada tugas 3 kali ini, ditambahkan lagi tiga konsep tambahan yaitu:

- Abstract class, sebagai parent class dari class Koin dan Monster

```
public abstract class KoinMonster {
    private int nilai;
    private String nama;

    /**
     *
     * @param nilai method constructor untuk menentukan nilai koin
     */
    public KoinMonster(int nilai) {
        this.nilai = nilai;
    }

    /**
     *
     * @param nilai method constructor untuk menentukan nama monster berserta nilai
     * pengurangannya.
     * @param nama
     */
    public KoinMonster(int nilai, String nama) {
        this.nilai = nilai;
        this.nama = nama;
    }

    /**
     *
     * @return method accessor untuk return dari nilai koin, serta juga dapat digunakan untuk return nilai monster.
     */
    public int getNilai() {
        return nilai;
    }

    /**
     *
     * @return Method accessor untuk return dari nama monster.
     */
    public String getName() {
        return this.nama;
    }

    /**
     *
     * @param nama Method mutator sebagai setter nama monster.
     */
    public void setName(String nama) {
        this.nama = nama;
    }

    /**
     *
     * @param nilaibaru method mutator sebagai setter nilai koin yang baru.
     */
    public void setNilai(int nilaibaru) {
        this.nilai = nilaibaru;
    }
}
```

- Interface

```

/*
 * Menyatakan class interface yang akan mengimplementasikan
 * method didalam class MainLompatKatak
 *
 * @author Furqan Ramadhan
 * @version 1.0
 */
public interface interfaceMainkan {
    void Mainkan();
}

```

- Inheritance, pada subclass Monster dan subclass Koin
class Koin

```

/*
 * Menyatakan class bernama Koin yang merupakan
 * subclass dari class abstract KoinMonster.
 *
 * @author Furqan Ramadhan
 * @version 1.2
 */
public class Koin extends KoinMonster {
    /**
     * @param method constructor serta memanggil
     *         class parent menggunakan keyword super()
     *         pada nilai.
     */
    public Koin(int nilai) {
        super(nilai: 10);
        // nilai = 10;
    }
}

```

class Monster

```

/**
 * Menyatakan class bernama Monster yang merupakan subclass dari
 * class abstract KoinMonster.
 *
 * @author Furqan Ramadhan
 * @version 1.4
 */
public class Monster extends KoinMonster {
    /**
     *
     * @param nilai constructor serta memanggil
     *         class parent menggunakan keyword super()
     *         pada nilai dan nama.
     * @param nama
     */
    public Monster(int nilai, String nama) {
        super(-8, nama: "Snake");
        /**
         * nilai = -8;
         * nama = "Snake";
         */
    }
}

```

Run Program

```

(furqan@ramadhan)-[/media/.../Semester 3/Pemrograman Berorientasi Objek/Blue-J/Tugas_03]
$ java MainLompatKatak
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Hai katak, melompatlah!...
Press any key to begin...

```

Insert nama player, serta posisi katak dan skor awal

```

Insert name player: cryptedlm
Player : cryptedlm
Choose your level..
1. Easy
2. Medium
3. Hard
2
Your level is : Medium
Position of cangkuk: 0
Score: 100
Do you want to go forward or backward? (a/d)

```

Katak maju 1 langkah

```
Do you want to go forward or backward? (a/d)
d
How much?
1. One step
2. Two steps
Insert your choice : 1
Position of cangguk: 1
Score: 100
```

Katak maju 2 langkah

```
Do you want to go forward or backward? (a/d)
d
How much?
1. One step
2. Two steps
Insert your choice : 2
Position of cangguk: 3
Score: 100
```

Katak mundur 1 langkah

```
Do you want to go forward or backward? (a/d)
a
How much?
1. One step
2. Two steps
Insert your choice : 1
Yuhuu, got a score!
Position of cangguk: 2
Score: 110
```

Katak mundur 2 langkah

```
How much?
1. One step
2. Two steps
Insert your choice : 2
Position of cangguk: 0
Score: 110
```

Katak kehabisan posisi dan exit(1)

```
How much?
1. One step
2. Two steps
Insert your choice : 1
You run out of positions, game over!
```

Satu langkah akhir

```
There is Monster!! Snake :) score reduced!  
Position of cangkuk: 498  
Score: 374  
One more jump.  
Do you want to go forward or backward? (a/d)
```

Akhir permainan

```
Do you want to go forward or backward? (a/d)  
d  
How much?  
1. One step  
2. Two steps  
Insert your choice : 1  
Yuhuu, got a score!  
All done!  
player : cryptedlm  
Score : 384  
Bad!
```

Statistik Akhir

```
player : cryptedlm  
Score : 384  
Bad!  
Play again ? (y/n) : n  
Thank you for playing!
```

Link video :

https://drive.google.com/file/d/1kTKx3f08K3Ggywy_oTlfxz81CQTAmEE6/view?usp=sharing