

Delmålskontroll, EDA501 Programmering

2015-02-27

Hjälpmedel: Java snabbreferens

Upplägg:

OBS! Den del av salen du sitter i kommer ha en egen övningsledare och du måste stanna i den salsdelen under alla momenten nedan!

Del 1: Du löser denna skrivning individuellt. Är du klar innan sluttiden får du lämna in och ta fram en bok, din telefon, eller liknande. Du får även gå ut en sväng om du vill – men var tyst så du inte stör de övriga skrivande.

Del 2: Ni sätter er parvis och får ut rättningsmallen som ni läser igenom. Efter ca 10 minuter får ni ut två andra personers skrivning som ni rättar enligt anvisningarna i rättningsmallen. När rättningstiden är slut samlar vi in alla rättade skrivningarna.

Del 3: Du får hämta din egen skrivning och titta på rättningen.

– Är du nöjd med rättningen lämnar du bara tillbaka skrivningen till din övningsledare igen.

– Är du inte nöjd med rättningen lämnar du skrivningen till Roy istället!

Sedan är det slut och fredagskväll☺

1. Skriv en klass med en main-metod som slår fem tärningslag med en tärning och skriver ut skillnaden mellan det högsta och lägsta tärningsslaget enligt exemplet nedan. För tärningsslagen ska den färdigskrivna klassen Die användas.

Exempel: Om de fem tärningsslagen är:

5 2 2 3 4

ska programmet ge följande utskrift:

Största differens: 3

Den färdigskrivna klassen Die har följande specifikation:

```
/** Skapar en tärning. */  
Die();  
  
/** Kastar tärningen. */  
void roll();  
  
/** Returnerar antal prickar tärningen visar. */  
int getResult();
```

VÄND!

2. Klassen Rook, vars specifikation finns nedan, beskriver ett schacktorn som förflyttar sig på ett schackbräde. Schackbrädet är ett vanligt schackbräde med 8x8 rutor, men där rutorna är numrerade med (x,y)-koordinaterna (1,1)..(8,8). Ett torn kan bara röra sig rakt horisontellt eller vertikalt med ett godtyckligt antal steg varje gång.

Specifikation av klassen Rook:

```
/** Skapar ett schacktorn som befinner sig i positionen (x,y)
    på ett schackbräde. */
Rook(int x, int y);

/** Tar reda på tornets position i X-led (horisontellt). */
int getX();

/** Tar reda på tornets position i Y-led (vertikalt). */
int getY();

/** Flyttar tornet n steg (n positivt eller negativt) i X-led.
    Du får förutsätta att tornet inte hamnar utanför brädet. */
void moveAlongX(int n);

/** Flyttar tornet n steg (n positivt eller negativt) i Y-led.
    Du får förutsätta att tornet inte hamnar utanför brädet. */
void moveAlongY(int n);

/** Avgör om det är tillåtet att flytta tornet till positionen (x,y).
    Du behöver inte ta hänsyn till andra pjäser på schackbrädet och
    du får förutsätta att (x,y) är inom brädet. */
boolean potentialPosition(int x, int y);
```

Implementera klassen Rook.

3. Studera följande program:

```
public class TestRook {
    public static void main(String[] args) {
        Rook r1 = new Rook(1,1);
        Rook r2 = r1;
        Rook r3 = new Rook(8,1);
        int nbr = 3;
        int step = nbr;
        for (int i=1; i<=nbr; i++) {
            r1.moveAlongY(step);
            step--;
        }

        System.out.println(r1.getX() + "," + r1.getY());
        System.out.println(r2.getX() + "," + r2.getY());
        System.out.println(r3.getX() + "," + r3.getY());
        ...
    }
}
```

- a) Hur ser utskriften ut från programmet?
- b) Vilka variabler deklarerar i main-metoden innan for-satsen?
- c) Hur många objekt skapas?
- d) Rita en figur som visar vilka variabler och objekt som finns efter det att satsen `System.out.println(r3.getX() + "," + r3.getY());` exekverats.

I figuren skall objektens attribut framgå. Ange också värdet av alla variabler och attribut. Rita referensvariabler med pilar på vanligt sätt.