

Tentamen i Imperativ Programmering med Grundläggande Objektorientering, DIT012

Joachim von Hacht

Datum: 2016-08-17

Tid: 08.30-12.30

Hjälpmedel: Engelskt-Valfritt språk lexikon

Betygsgränser:

- U: -23
- G: 24-43
- VG: 44-60 (max 60)

Lärare: Joachim von Hacht. Någon besöker ca 10.00 och 11.30, tel. 0707/311066

Granskning: Tentamen kan granskas på studieexpeditionen. Vi ev. åsikter om rättningen eposta mig och ange noggrant vad du anser är fel så återkommer jag (ta en bild och skicka).

Instruktioner:

- För full poäng på essä-frågor krävs ett läsbart, begripligt och heltäckande svar. Generellt 1p för varje relevant aspekt av problemet. Oprecisa eller alltför generella (vaga) svar ger inga poäng. Konkretisera och/eller ge exempel. Det är aldrig någon risk att vara övertydlig!
- Det räcker med enbart relevanta kodavsnitt, övrig kod ersätts med “...” (aldrig import, main-metod, etc....)
- Överkomplicerade lösningar kan ge poängavdrag.
- Vi utgår från att användaren alltid skriver rätt och/eller gör rätt (d.v.s ingen felhantering behövs). Om felhantering skall ingå anges detta specifikt.

LYCKA TILL...

1. Förklara med en eller ett par meningar: 2p
- a) Vad avses med en referensvariabel?
- b) Vad är vitsen med Javas typsystem (varför används ett typsystem)?

2. Skriv ett program som löser ett ekvationssystem av typen $\begin{cases} ax + by = e \\ cx + dy = f \end{cases}$.

Lösningen ges av $x = \frac{ed-bf}{ad-bc}$ och $y = \frac{af-ec}{ad-bc}$. Vi tar ingen hänsyn till om nämnarna blir 0. Programmet frågar efter alla koefficienter och skriver därefter ut värdet för x och y. Exempelkörning:

3p

```
Input a, b and e for ax + by = e > 2.5 4 6
Input c, d and f for cx + dy = f > 3 6 7.7
x = 1.7333333333333332 y = 0.4166666666666667
```

3. Skriv en metod isPlaindromNumber(int n). Metoden avgör om ett heltal ger samma värde om man vänder på siffrorna. Om så returneras true annars false. Exempel: 4p

```
out.println(isPalindromNumber(12321));    // true
out.println(isPalindromNumber(9876789));  // true
out.println(isPalindromNumber(123123));   // false
```

4. Beskriv och illustrera med kod, i detalj, hur ett metodanrop i Java går till. OBS! För full poäng måste varje steg i förloppet tydligt redovisas. 6p

5. En "klump" är en del av en heltals-array där alla element har samma värde. D.v.s. minst två intilliggande index måste ha samma värde. Exempel: 6p

```
{1, 2, 3, 4, 5}           // Ingen klump
{1, 1, 3, 4, 5, 6, 8, 9}   // En klump 1,1
{1, 1, 3, 4, 5, 8, 8, 8}   // Två klumpar 1,1 och 8,8,8
{5, 5, 4, 4, 3, 8, 8, 8}   // Tre klumpar 5,5 och 4,4 och 8,8,8
{1, 1, 1, 1}              // En klump
```

Skriv en metod som givet ett fält returnerar antalet klumpar i fältet.

6. Skriv en metod som givet en heltalsmatris ($\geq 2 \times 2$ rader och kolumner) returnerar det största värdet av någon 2×2 delmatris. Exempel: 8p

```
1 0 1
2 2 0      Största värdet ges av delmatris 2 2, värdet är 8.
1 3 0                                1 3
```

7. En NPC (Non-Player character) är en karaktär som används i spel, ofta som fiender. En NPC styrs av datorn m. h. a. artificiell intelligens, därav namnet non-player. Skapa en klass för NPC-objekt. Följande gäller:

7p

- a) En NPC har position x och y samt hälsa och kraft.
- b) Klassen skall ha en konstruktor som initierar all data.
- c) Klassen skall ha en kopieringskonstruktor.
- d) Klassen skall ha en metod boolean isDead().
- e) Visa hur man skapar en instans av klassen och hur man avgör om en NPC är död.

8. Skapa en metod som vänder på en text. Exempelvis blir "The fox jumps over the lazy dog" till "dog lazy the over jumps fox The". OBS! Inget får tillkomma eller försvinna i den vända texten. Följande får användas:

6p

```
// Metoder från String-klassen
- charAt(int i), ger tecknet vid index i.
- indexOf(char ch), ger index för tecknet ch, -1 om tecknet saknas.
- length() ger längden av strängen.
- substring(int start, int end), ger en delsträng från
  start (inkl.) till end-1.
- substring(int start), ger en delsträng från start (inkl.)
  till strängens slut.
- split(String str), delar upp en sträng i en array av delsträngar
  utifrån ett visst tecken. Returnerar en String-array (String[])
  Exempel "aaa:bb:cccc:dd".split(":") -> [ "aaa", "bb", "cccc", "dd" ]
// Metoder ur StringBuilder-klassen
- append(), lägger till tecken i slutet på en text.
```

9. Spelet fyra i rad (Connect Four) spelas av två spelare med olika färgade brickor. Spelarna placerar växelvis brickor i en ram genom att släppa dem genom en öppning i spelplanens överkant. Brickorna faller nedåt till dess de når underkanten eller till dess de stoppas av en tidigare placerad bricka. Brickorna faller rakt ner, de kan inte byta "stapel". Den spelare som först fått 4 brickor i rad (samma färg) vinner. Se bild.

8p



Figure 1: Spelet "Fyra i rad"

Skriv en klass som kan representera spelet. Klassen skall använda den färdiga Java-klassen `Color` för att representera brickorna. Du väljer själv hur ramen skall representeras. Klassen skall ha en metod `boolean put(int col, Color c)` som placerar en bricka med given färg i en viss vertikal rad (kolumn) i ramen. Om detta lyckades är resultatet `true` annars `false`.

10. Vad kommer programmet nedan att skriva ut? Förklara utförligt varför resultatet blir som det blir! Förklara varje steg som leder till resultatet. `Calendar` är en färdig Java-klass som vid anrop av metoden `get()` ger aktuellt år (2016).

10p

```
public class Elvis {

    public static final Elvis INSTANCE = new Elvis();
    private final int beltSize;
    private static final int CURRENT_YEAR =
        Calendar.getInstance().get(Calendar.YEAR);
    private Elvis() {
        beltSize = CURRENT_YEAR - 1930;
    }
    public int beltSize() {
        return beltSize;
    }
    public static void main(String[] args) {
        System.out.println("Elvis wears a size "
            + INSTANCE.beltSize() + " belt.");
    }
}
```