

Tentamen Programmeren in Java

16 december 2009

Voorzie elk blad dat je inlevert duidelijk van je naam en collegekaartnummer. Schrijf netjes, met hele zinnen, zonder spelfouten.

Opgaven 1 t/m 5 dienen gemaakt te worden door diegenen die voor het deeltentamen in oktober een onvoldoende (< 5.5) hadden. Opgaven 6 t/m 10 zijn voor iedereen.

1. Geef aan of er fouten (i.e., fouten die bij compilatie aan het licht komen) in de volgende statements zitten, en zo ja, leg uit wat die fouten zijn:

- (a) `char "x" = (char) 2;`
- (b) `byte a = 255, b = --a;`
- (c) `float f = (1.0 + 1);`
- (d) `double z = 1F - 2 * (int)(1.0);`
- (e) `while (true);`
- (f) `boolean h = (boolean) 1;`
- (g) `byte B = (byte)(300);`
- (h) `boolean p = ! true;`

2. Wat zijn de waarden van de variabelen na uitvoering van de assignments?

- (a) `int a1 = 3, a2 = 4; a2 *= a1++;`
- (b) `byte b = 9 / 2 * 2;`
- (c) `double c = 30 / 4 + (5 * 9);`
- (d) `int d1 = 3, d2 = ((--d1 == 3) ? --d1 : 4);`
- (e) `double e = -5.0; e -= 2 - 2 * e;`
- (f) `String s = 1 + 2 + "1";`
- (g) `double g = -1 - Byte.MIN.VALUE;`
- (h) `int h = 2 * Byte.MAX.VALUE;`

3. Beschouw de volgende code:

```
public class Opgave3 {
    public static void main(String[] args){

        int i = 0;
        while (i < 5) {
            for (int j = i; j > 1; j--)
                System.out.println(j + " ");
            System.out.println("****");
            i++;
        }
    }
}
```

Wat is de uitvoer van dit programma?

4. Beschouw de volgende code:

```
public class Opgave4 {
    public static void main(String[] args){
        int number = 0;
        int[] numbers = new int[1];

        m(number, numbers);
        System.out.println("number is " + number);
        System.out.println("numbers is " + numbers[0]);
    }
    public static void m(int x, int[] y) {
        x = 3;
        y[0] = 3;
    }
}
```

- Laat de uitvoer van het programma zien.
- Wat is het verschil tussen primitieve variabelen en referentie variabelen?
- Wat is het verschil tussen call by reference en call by value?

5. (a) Beschrijf het verschil tussen static variabelen en instantiatievariabelen.
(b) Laat de uitvoer van het volgende programma zien.

```
public class Opgave5 {
    static int i = 0;
    static int j = 0;
    public static void main(String[] args) {
        int i = 2;
        int k = 3;
        {
            int j = 3;
            System.out.println("i + j is " + i + j);
        }

        k = i + j;
        System.out.println("k is " + k);
        System.out.println("j is " + j);
    }
}
```

6. Beschouw de volgende code.

```
public class Opgave6 {
    public static void main(String[] args) {
        print(new Kind());
        print(new Vader());
        print(new GrootVader());
        print(new Object());
    }

    public static void print(Object x) {
        System.out.println(x.toString());
    }
}

class GrootVader extends Object {
    public String toString() {
        return "grootvader";
    }
}

class Vader extends GrootVader {
    public String toString() {
        return "vader";
    }
}

class Kind extends Vader {
}
```

- (a) Wat is de uitvoer van dit programma?

(b) Wat is dynamic binding?

(c) Leg uit hoe dynamic binding werkt in dit programma?

7. (a) Zijn alle aanroepen in het volgende programma toegestaan?

(b) Wat is de uitvoer als de probleemregel(s) verwijderd is/zijn?

```
public class Opgave7 {
    public static void main(String [] args){
        C c = new C(3, 4);
        System.out.println(c.m1());
        System.out.println(c.m2());
        System.out.println(c.m3());
        System.out.println(c.m4());
        P q = c;
        System.out.println(q.m1());
        System.out.println(q.m2());
        System.out.println(q.m3());
        System.out.println(q.m4());
    }
}
class P {
    int pn;
    P(int pn) {this.pn = pn;}

    int m1() {return(pn);}
    int m3() {return(pn * pn);}
    static int m4() {return 4;}
}
class C extends P {
    int cn;
    C(int pn, int cn) {super(pn); this.cn = cn;}

    int m2() {return(pn + cn);}
    int m3() {return(pn * cn);}
    static int m4() {return 5;}
}
```

8. Gegeven het volgende programma.

```
public class TelOp {
    public static void main(String [] args) {
        try {
            int som = Integer.parseInt(args[0]);
            for (int i = 1; i < args.length; i++) {
                som = som + Integer.parseInt(args[i]);
            }
            System.out.println(som);
        }
        catch (Exception ex) {
            System.out.println("Er gaat iets fout");
        }
        System.out.println("Einde");
    }
}
```

10. (a) Wat is het verschil tussen overladen en overriden van methoden?
(b) De `equals` methode uit de `Object` klasse kan overriden worden. Doe dat voor de volgende `Cirkel` klasse

```
public class Opgave10 {
    public static void main(String[] args) {
        Object object1 = new Cirkel(2);
        Object object2 = new Cirkel(2);

        System.out.println(object1.equals(object2));
    }
}

class Cirkel {
    double r;
    Cirkel(double straal) {
        r = straal;
    }
    ...
}
```