

Opgave 1.

- a) Gegeven zijn de volgende classes

```
class F {
    int i;

    F (int i) {
        this.i = i;
    }

    F () {
        this(0);
    }

    ....
}
```

```
class G extends F {
    int j;
}
```

- Schrijf voor de class G een default constructor die de i uit de class F de waarde 3 geeft en de j uit de class G de waarde 5 geeft.

Aan de class G wordt nu de volgende declaratie toegevoegd

```
int i;
```

- Schrijf voor de class G een default constructor die de i uit de class F de waarde 3 geeft, de j uit de class G de waarde 5 geeft en de i uit de class G de waarde 7 geeft.

De declaratie van de variabele i in de class F wordt gewijzigd in

```
private int i;
```

- Schrijf nu weer voor de class G een default constructor die de i uit de class F de waarde 3 geeft, de j uit de class G de waarde 5 geeft en de i uit de class G de waarde 7 geeft.

- b) Wat is auto boxing en auto unboxing? Geef van beide een voorbeeld.
c) Wat is het effect van de modifier private op variabelen?

Kunnen de methodes concat() in de onderstaande classes M en P beschikken over de verschillende private variabelen die ze willen gebruiken?

```
class M {
    private String s;

    .....

    void concat (M m) {
        s = s + m.s;
    }

    .....
}
```

```
class P {
    private String s;

    .....

    void concat (M m) {
        s = s + m.s;
    }

    .....
}
```

- d) Gegeven is de volgende declaratie

```
PrintStream out = new PrintStream(System.out);
```

Schrijf een methode print(), die kan worden aangeroepen met een willekeurig aantal argumenten van het type int, die de waarden van deze argumenten ieder op een aparte regel afdruckt. Gebruik hierbij het for-each statement.

- e) - Wat is een abstracte methode?
- Wat is een abstracte class?
- Wanneer is het verplicht om een class abstract te maken?
- Wat is het nut om in een abstracte class een constructor te hebben?
- Is het mogelijk dat een abstracte class geen abstracte methodes bevat?
- Abstracte methodes kunnen in abstracte classes staan. Waarin kunnen ze nog meer staan?

Opgave 2.

- a) Maak voor een Rij-ADT een specificatie van een geparametriseerd (generic) Rij-type met een default constructor, een methode remove() die het element op een gegeven index-positie verwijdert, een methode size() die het aantal elementen retourneert en een methode contains() die voor een gegeven waarde antwoord geeft op de vraag of deze waarde in de rij zit.

De type-parameters moeten beperkt (bound) worden tot objecten die het type Comparable hebben.

Specificeer het data-type m.b.v. de eigenschappen elementen, structuur en domein. Specificeer de operaties m.b.v. PRE- en POST-conditie.

Specificeer de rij zo dat het eerste element de index 0 heeft.

- b) Maak een geparametriseerde class Knoop welke een variabele data en een variabele next bevat. Het type van data is het als type-parameter meegegeven type. De variabele next moet een zodanig type hebben dat next een volgende knoop van het type Knoop kan aanwijzen. Voor deze opgave hoeft de class Knoop verder alleen een constructor te bevatten waaraan bij aanroep als argument de waarde van data meegegeven wordt.
c) Maak voor het Rij-ADT een implementatie op basis van de bij onderdeel a) gemaakte specificatie. Gebruik een lijst van knopen van het type Knoop uit onderdeel b) om de datastructuur te realiseren. Het is niet nodig om een aparte Lijst-class te maken.

Beoordelingstabel	opgave	a	b	c	d	e	totaal
	1	3	3	3	3	4	16
	2	9	2	9			20
	TOTAAL						36

Het eindcijfer wordt als volgt uit het totaal T verkregen : $E = T/4 + 1$.