

Opgave 1.

- a) Wat is auto boxing en auto unboxing? Geef van beide een voorbeeld.
- b) Bij overerving (inheritance) van een class X in een class Y, erft (inherit) de class Y velden (members) van de class X. Niet alle velden worden echter geerfd. Geef aan welke velden uit de class X niet geerfd worden in de subclass Y.
- Is het mogelijk om in een superclass te definiëren dat subclasses een bepaalde methode uit de superclass niet zullen erven? Zo ja, hoe dan?
- Is het mogelijk om in een subclass een methode, die uit de superclass geerfd wordt, te herdefinieren? Zo ja, hoe dan?
- Is het mogelijk in een superclass te definiëren dat subclasses een bepaalde methode, die ze van de superclass erven, niet mogen herdefinieren? Zo ja, hoe dan?
- c) Bespreek het verschil tussen `super()` en `super` door uit te leggen wanneer ieder van beide gebruikt wordt.
- d) Schrijf een functie `som()`, die kan worden aangeroepen met een willekeurig aantal argumenten van het type `String`, die de concatenatie van de waarden van deze argumenten bepaalt en retourneert. Gebruik hierbij het `for-each` statement.
- e) Gegeven is een generieke class Y met 1 variabele data

```
class Y<T> {  
    private ArrayList<T> data;  
  
    .....  
}
```

Verder is gegeven dat de interface `Iterable` er als volgt uitziet

```
interface Iterable<T> {  
    Iterator<T> iterator();  
}
```

Geef de code die aan de class `Y<T>` moet worden toegevoegd indien deze de interface `Iterable<T>` gaat implementeren, m.a.w. als `"class Y<T> {"` verandert in `"class Y<T> implements Iterable<T> {"`.

Verder is gegeven dat in een programma dat de gewijzigde class `Y<T>` gebruikt de onderstaande declaratie van een variabele `y` staat

```
Y<String> y;
```

Ten slotte is gegeven dat de variabele `y` een waarde gekregen heeft.

Schrijf een stukje code waarmee in het programma de in de variabele `y` opgeslagen `String`'s afgedrukt kunnen worden. Druk iedere `String` op een aparte regel af.

Opgave 2.

- a) Gegeven is de interface

```
interface Kopieerbaar extends Cloneable {  
    public Object clone();  
}
```

Maak voor een Queue-ADT een specificatie van een geparametriseerd (generic) Queue-type met een default constructor, een methode `enqueue()` die een kopie van een gegeven element achteraan de queue toevoegt, een methode `size` die het aantal elementen retourneert, een methode `dequeue()` die het element vooraan in de queue verwijdert en retourneert en een methode `front()` die een kopie van het element vooraan in de queue retourneert.

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

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

- b) Maak voor het Queue-ADT een implementatie op basis van een array. Doe dit op zo'n manier dat het mogelijk is om een willekeurig aantal elementen op te slaan.

Beoordelingstabel

opgave	a	b	c	d	e	totaal
1	3	4	2	3	4	16
2	8	12				20
TOTAAL						36

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