

**Tentamen Inleiding Programmeren (IN1608WI), 29 januari 2008 14.00-16.00,
Technische Universiteit Delft, Faculteit EWI, Afdeling 2.**

Het tentamen bestaat uit 4 opgaven. Je mag gebruik maken van het studieboek.

Opgave 1(18 punten).

In Bijlage 1 vind je de definitie van de klassen Item en Telefoonboek.

Bestudeer deze klassen goed voordat je deze vraag beantwoordt.

In de klasse Telefoonboek is methode test gedefinieerd. Wat is na uitvoering van deze methode de waarde van elk van de volgende expressies? **Licht je antwoord toe!**

Mocht een van de expressies aanleiding geven tot een compileer- of run time-fout, dan dien je dat aan te geven samen met de reden van de fout.

- a. `items[0] == items[2]`
- b. `items[1] == items[2]`
- c. `items[0].getNaam()`
- d. `items[1].getNaam()`
- e. `items[2].getNaam() == items[0].getNaam()`
- f. `items[2].getNummer() == items[1].getNummer()`

Opgave 2(24 punten).

Aan de klasse Telefoonboek (uit Bijlage 1) wordt een drietal methoden toegevoegd.

Gevraagd wordt om een implementatie van de volgende methoden. Het is niet toegestaan om binnen een for- of een while-opdracht een return-opdracht te gebruiken.

- a. `indexVan(naam : String) : int`
post: geeft de index terug van het Item waarvan de naam gelijk is aan de opgegeven naam. Als de naam niet voorkomt wordt -1 teruggegeven.
- b. `voegtoe(ti : Item)`
post: Als ti.naam al voorkwam in het telefoonboek, is het nummer behorend bij deze naam vervangen door ti.nummer, zo niet, dan is item ti als laatste element aan het telefoonboek toegevoegd.
- c. `zoekNummer(naam: String) : String`
post: het nummer behorende bij deze naam. Als de naam niet voorkomt, wordt een lege String teruggegeven.

Opgave 3(24 punten).

In Bijlage 2 vind je de definitie van een interface en aantal klassen. Bestudeer deze goed voordat je de vragen beantwoordt.

- a. Geef een klassendiagram waarin de relaties tussen het interface en de klassen is weergegeven.
 - b. Hieronder zijn 4 stukjes code gegeven die zijn gebaseerd op de definities uit Bijlage 2. Geef van elk stukje code aan
 - of er een fout optreedt tijdens het vertalen
 - (als er geen vertaalfout optreedt) of er tijdens de uitvoering van de code een fout optreedt, en zo ja welke,
 - (als er geen vertaal- of uitvoeringsfout optreedt) wat de waarde van c is na uitvoering
1.

```
Code code = new Code();  
char c = code.encode('A');
```
 2.

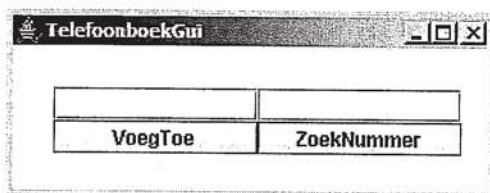
```
Code code = new SimpleCode();  
char c = code.encode('B');
```
 3.

```
SimpleCode code = new SmartCode(2);  
char c = code.encode('C');
```
 4.

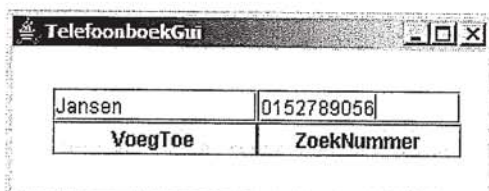
```
SimpleCode code = new SmartCode(2);  
int k = code.getKey();  
char c = code.encode('p');
```

Opgave 4(24 punten).

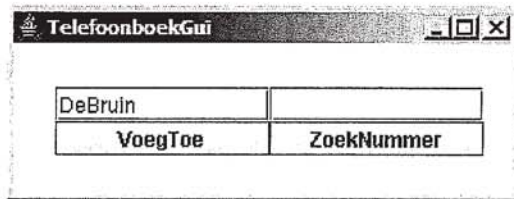
In Bijlage 3 vind je de implementatie van het grafische userinterface dat hieronder is afgebeeld. Het userinterface maakt gebruik van een Telefoonboek-object.



Het userinterface heeft twee knoppen. Met de eerste knop (VoegToe) worden een naam en nummer samengevoegd tot een Item dat aan het Telefoonboek-object wordt toegevoegd. De naam en nummer worden ingevoerd via de twee tekstvelden. Na invoer worden de tekstvelden leeg gemaakt. Zie de figuur hieronder.



Met de tweede knop (ZoekNummer) kan het nummer worden gezocht behorend bij een gegeven naam. De naam wordt ingevoerd via het naam tekstveld. Het zoekresultaat wordt getoond in het nummer tekstveld. Zie de figuur hieronder.



De implementatie is nog niet volledig. Vul de code van Bijlage 3 zo aan dat het grafische userinterface werkt op de manier zoals hierboven is beschreven.

Bijlage 1.

```
public class Item {
    private String naam;
    private String nummer;

    public Item(String na, String nu){
        setNaam(na);
        setNummer(nu);
    }

    public String getNaam(){          return naam;      }
    public String getNummer(){        return nummer;    }
    public void setNaam(String na){   naam = na;      }
    public void setNummer(String nu){ nummer = nu;      }
}
```

```
public class Telefoonboek{
    private Item[] items;
    private int aantal;

    public Telefoonboek(){
        items = new Item[1000];
        aantal = 0;
    }

    public void test(){
        Item item = new Item("Jansen","314159265");
        items[aantal] = item;
        aantal++;

        item = new Item("DeBruin","271828182");
        items[aantal] = item;
        aantal++;

        item.setNaam(items[0].getNaam());
        item.setNummer(items[0].getNummer());
        items[aantal] = item;
        aantal++;
    }
}
```