Departement Informatica en Informatiekunde, Faculteit Bètawetenschappen, UU.
In elektronische vorm beschikbaar gemaakt door de TBC van A–Eskwadraat.
Het college INFOIMP werd in 2006/2007 gegeven door Jeroen Fokker.

Imperatief programmeren, tweede deeltentamen (INFOIMP) 13 oktober 2006

Opgave 1:

In de klasse `String` zit onder andere twee methodes met de naam `indexOf`, die verschillen in het type parameter dat ze krijgen:

```
int indexOf(char c)
int indexOf(String s)
```

Deze eerste methode levert het nummer van de eerste positie op waar `c` in de string voorkomt. Als `c` nergens in de string voorkomt, is het resultaat 1.
Voorbeelden:

`"Utrecht".indexOf('h')`	geef 5
`"Utrecht".indexOf('t')`	geef 1
`"Utrecht".indexOf('x')`	geef -1

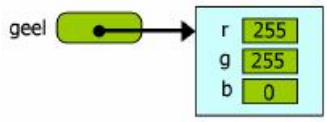
De tweede methode (die niet in de samenvatting staat, maar die wel bestaat) levert de eerste plaats waar `s` een deel is van het totaal (of -1 als `s` nergens voorkomt).
Voorbeelden:

`"Utrecht".indexOf('ech')`	geef 5
`"Utrecht".indexOf('trh')`	geef 1
`"Utrecht".indexOf('Utrecht')`	geef -1

Stel dat je de auteur van de klasse `String` bent. Alle andere methoden van die klasse zijn al geschreven, behalve de `subString` en `indexOf` methoden.
De opgave: Schrijf de twee hierboven beschreven `indexOf` methoden. (Het is toegestaan, maar niet verplicht, om extra hulp-methodes te schrijven).

Opgave 2

Het plaatje hieronder toont de situatie in het geheugen die ontstaat na uitvoering van
`Color geel = new Color(255,255,0);`



Gegeven zijn de volgende klasse-definities:

```
class Een
{   int x;
    public Een()
    {   x = 0;
    }
    public void setX(int a)
    {   x = a;
    }
}
class Twee
{   int x; Een e;
    public Twee(Een b, int c)
    {   e = b;
        x = c+1
    }
    public Een getE()
    {   return e;
    }
}
class Drie extends Een
{   Twee p, q;
    public Drie()
    {   p = new Twee( new Een(),1 );
        p.getE().setX(7);
        q = new Twee( p.getE(), 2 );
        q.getE().setX(!);
        p = new Twee( this,     1 );
        p.getE().setX(7);
    }
}
```

Teken, in dezelfde stijl als het voorbeeldplaatje, de situatie die in het geheugen ontstaat na uitvoering van
`Drie d = new Drie();`
Maak, net als in het voorbeeld, duidelijk onderscheid tussen de naam en de waarde van de variabalen: de naam staat *naast* de hokjes, de waarde er *in*. Object-verwijzingen moeten, net als in het voorbeeld, met een duidelijke stip beginnen *in* het hokje van de verwijzings-variabele, en wijzen naar de *rand* van het object.

Opgave 3:

a) Bekijk het programma-fragment in het kader. Geef voor elk van de zes variabelen x1 t/m x6 aan welk van de volgende vier gevallen geldt:

- sub T: het type van x is klasse T of een subklasse daarvan
- super T: het type van x is klasse T of een superklasse daarvan
- impl T: het type van x is een klasse die T implementeert
- prim T: het type van x is het primitieve type T

waarbij je ook steeds een keuze voor T maakt.

*Het antwoord op dit onderdeel heeft dus een vaste vorm: voor elk van de zes variabelen x1 t/m x6 het woord **sub**, **super**, **impl** of **prim**, met daarachter de naam van de klasse, interface, of type die je kiest. Dus zes regels zoals:*

x7: sub Color

x8: super String

x9: impl Map

Een toelichting is niet nodig.

```
Button b;
b = new Button("hoi");
b.addActionListener(x1);

x2.add(x3)

x4 = Font.BOLD;

x5 = new TextField(10);

Thread t;
t = new Thread(x6);
```

(Van de vele methoden in Java die `add` heten, is hier de methode bedoeld waarmee in AWT een grafische userinterface wordt opgebouwd.)

b) Wat verandert er in het object `b` en/of het object `x1` door de aanroep van `b.addActionListener(x1);`? Waarom is dat nodig?

c) Hoe zal het object `t` later in het programma waarschijnlijk gebruikt worden? Wat gebeurt er op dat moment met het object `t` en/of het object `x6`?
Wat is daaraan zo bijzonder?

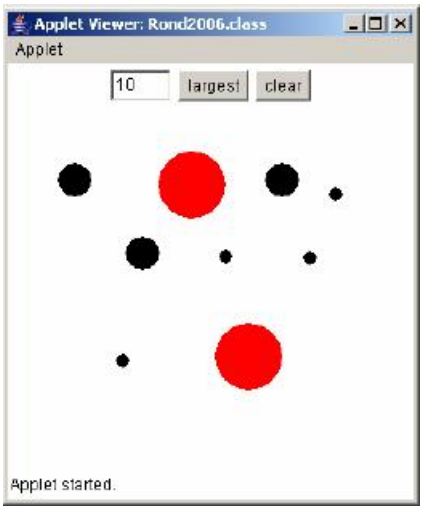
Opgave 4:

Om het schrijfwerk te beperken mag je in deze opgave:

- De HTML-file weglaten: je hoeft alleen de Java-file te schrijven
- De `import`-regels bovenaan het programma weglaten
- De muis-methoden die een lege body hebben weglaten

Het is niet erg als er in dit programma door afrondfouten kleine afwijkingen ontstaan.

Schrijf een applet met de volgende eigenschappen:



1. De gebruiker ziet bovenaan het scherm:
   - een tekstvak om een getal in te voeren (aan het begin is hier 25 ingevuld)
   - button met opschrift "largest" en een met opschrift "clear"
2. Iedere keer als de gebruiker ergens klikt ontstaat er, gecentreerd op dat punt, een cirkel met als diameter het getal in het tekstvlak op het moment van klikken.
3. Er zijn maximaal 100 cirkels zichtbaar. Als de gebruiker daarna toch meer punten aanklikt, gebeurt er niets (maar er mag ook geen run-time fout optreden!).
4. Na het indrukken van de "largest" worden de cirkel(s) met de grootste diameter rood.
5. Na het nogmaals indrukken van "largest" worden alle cirkels weer zwart, bij de derde klik verschijnt de kleur weer, enz.
6. Na het indrukken van "clear" verdwijnen alle cirkels.
   De gebruiker kan dan weer met 100 nieuwe kliks beginnen.