

**Tentamen Inleiding programmeren (IN1608WI), 30 januari 2006, 14.00-16.00,
Technische Universiteit Delft, Faculteit EWI, Afdeling 2.**

Bij dit tentamen mag je gebruik maken van “Java, Program Design” van Cohoon en Davidson of van “Java, First Contact” van Garside en Mariani.

Opgave 1 (25 punten).

Gegeven de specificatie van klasse Point waarmee een positie in R^2 wordt vastgelegd.

Point
-x: int -y: int
+Point(xx : int, yy : int) //post: has set x to xx and y to yy +getX(): int //post: returns x +setX(xx : int) //post: has set x to xx +getY(): int //post: returns y +setY(yy : int) //post: has set y to yy

Voor een project is een klasse nodig die lijkt op klasse Point. Bij de benodigde klasse zijn de waarden van x en y beperkt tot het interval -100,...,+100. Als er waarden buiten dit interval worden gebruikt, wordt er een exception geworpen. Hieronder is een deel van de benodigde klasse (met de naam RobustPoint) gegeven.

```
RobustPoint
+MIN = -100
+MAX = +100
+RobustPoint(xx : int, yy : int)
//pre: MIN <= xx <= MAX, MIN <= yy <= MAX
//post: if pre, has set x to xx and y to yy,
//      otherwise an exception has been thrown

+setX(xx : int)
//pre: MIN <= xx <= MAX
//post: if pre, has set x to xx,
//      otherwise an exception has been thrown

+setY(yy : int)
//pre: MIN <= yy <= MAX
//post: if pre, has set y to yy,
//      otherwise an exception has been thrown

+toString(): String
//post: return String representation of RobustPoint

+equals(other: Object): boolean
//pre: other is a Point-object
//post: returns true iff coordinates of this
//      and other are equal

+lessThan(other: Object): boolean
//pre: other is a Point-object
//post: returns true iff modulus of this
//      is less than modulus of other
```

De modulus is de afstand van het punt tot de oorsprong ($\sqrt{x^2 + y^2}$)

- a. Geef een implementatie van klasse RobustPoint. Gebruik bij de implementatie volgende aanwijzingen:

Maak bij de implementatie (via overerving) gebruik van klasse Point.
 Implementeer MIN en MAX als constanten.
 Gebruik klasse RuntimeException voor het werpen van de excepties.

- b. Gegeven het volgende programma waarin enkele objecten worden gecreëerd:

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

    RobustPoint p1 = new RobustPoint(10,10);
    Point p2 = new Point(12,8);

    System.out.println(p1);
    System.out.println(p1.lessThan(p2));

    RobustPoint p3 = new RobustPoint(-1000,0);
}
```

Leg uit wat er wordt afgedrukt op het scherm als het programma wordt uitgevoerd.

Opgave 2(25 punten)

Gegeven de specificatie van klasse PointList, waarmee een lijst van maximaal n Point-objecten kan worden gemaakt. De lijst bevat alleen unieke objecten, d.w.z. een object kan maar een keer in de lijst worden opgeslagen. De objecten in de lijst vormen een aaneengesloten rij.

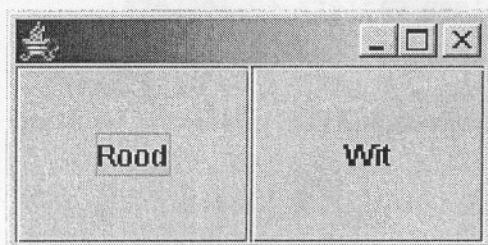
PointList
<pre>-points: Point[] -size: int //het aantal Point-objecten in de lijst</pre>
<pre>+PointList(n: int) //post: has constructed an empty PointList with a capacity of n +add(p: Point) //pre: size < capacity, the list doesn't contain a Point equal to p //post: iff pre, has added p to the list +contains(p: Point): boolean //post: returns true iff the list contains a Point equal to p +remove(p: Point) //post: iff list contained a Point-object equal to p, then this // object has been removed +toString(): String //post: returns a String representation of the list</pre>

Geef een implementatie van klasse PointList.

Opgave 3(25 punten)

Hieronder zie je de de grafische user interface van een eenvoudig programma. In de bijlage is de volledige code van dit programma vermeld.

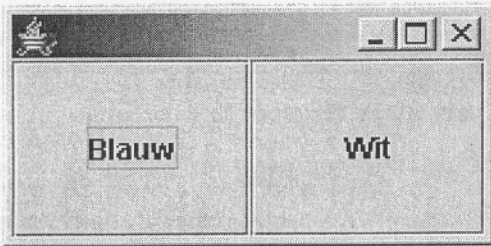
De user interface bevat twee knoppen (buttonLeft en buttonRight) die elk een opschrift hebben. Als er op een van de knoppen wordt geklikt, gebeurt er niets.



Gevraagd wordt om een uitbreiding van de code van het programma.

Na de uitbreiding moet het volgende gelden: het programma bevat 3 Strings: "Rood", "Wit" of "Blauw". Twee van de Strings zijn zichtbaar als opschrift van de knoppen. Wordt er op een van de knoppen geklikt, dan dient het opschrift van de knop te veranderen in de derde (niet zichtbare) String.

Dus als op de linkerknop wordt geklikt ziet de GUI er daarna als volgt uit:



Maak de bijlage los van het tentamen, schrijf je naam en studienummer erop. Geef in de bijlage aan welke veranderingen en uitbreidingen er nodig zijn om het programma op de gewenste manier te laten werken.

Er is een methode `derdeTekst()` beschikbaar, waarmee je de derde String kunt berekenen. Vb. `derdeTekst("Blauw", "Wit") -> "Rood"`. Je hoeft deze methode niet te implementeren.

Opgave 4 (15 punten)

Gegeven een string die uitsluitend letters en cijfers bevat. Gevraagd wordt een **recursieve** methode `aantalCijfers()` die de cijfers in de string telt. Je moet gebruik maken van recursie, een oplossing die gebruik van een herhalingsopdracht krijgt 0 punten!

```
public static int aantalCijfers(String s, int i )
//pre:  s is een niet lege String met alleen cijfers en letters,
//      0 <= i <= s.length()
//post: retourneert het aantal cijfers dat in s voorkomt
```

Je kunt bij de implementatie gebruik maken van methode `isDigit()` uit klasse `Character`, die aangeeft of een char een cijfer is of niet.

```
Character.isDigit(c : char) : boolean
```

Toelichting:

De waarde van `aantalCijfers("a1a1a1",0) -> 3`
de waarde van `aantalCijfers("11",0) -> 2`
en van `aantalCijfers("aaa",0) -> 0`.