

## Imperatief Programmeren (INFOIMP)

### 14 oktober 2009

#### Opgave 1.

In de klasse `String` zit een methode `replace`. Deze methode levert een nieuwe string op, waarin elk voorkomen van het character dat als eerste parameter wordt meegegeven, is vervangen door het character dat als tweede parameter wordt meegegeven. Bijvoorbeeld:

```
"Utrecht".replace('t','x') geeft "Uxrechx"  
"A+2+#@?".replace('+','9') geeft "A929#@?"
```

Ook is er een methode `endsWith`, die oplevert of een string eindigt met de string die als parameter wordt meegegeven. Bijvoorbeeld:

```
"Utrecht".endsWith("recht") geeft true
```

Stel dat je de auteur van de klasse `String` bent. Veel andere methoden van die klasse zijn al geschreven (die mag je dus gebruiken), maar nog niet de `substring` en `indexOf` methoden (die mag je dus gebruiken).

*De opgave:*

- Schrijf de methode `replace`.
- Schrijf de methode `endsWith`.
- In de klasse `String` is er ook een methode `equals`.  
In welke situatie geeft een aanroep van `equals` een ander resultaat dan het gebruik van de operator `==` ?

*(Het is toegestaan, maar niet verplicht, om extra hulp-methodes te schrijven).*

#### Opgave 2.

- Bekijk het programma-fragment in het kader. Geef voor elk van de zes variabelen `x1` t/m `x6` aan welk van de volgende vier getallen geldt:
  - `sub T`: het type van `x` is klasse `T` of een subklasse daarvan
  - `superT`: 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**, **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.*

```

this.addMouseListener(x1);

x2 = Scrollbar.HORIZONTAL;

x3 = new TextArea(10, 5);
x3 = new TextField(10);

x4.add(x5);

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 `this` en/of het object `x1` door de aanroep van `this.addMouseListener(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 3.

Om het schrijfwerk te beperken mag je in deze opgave:

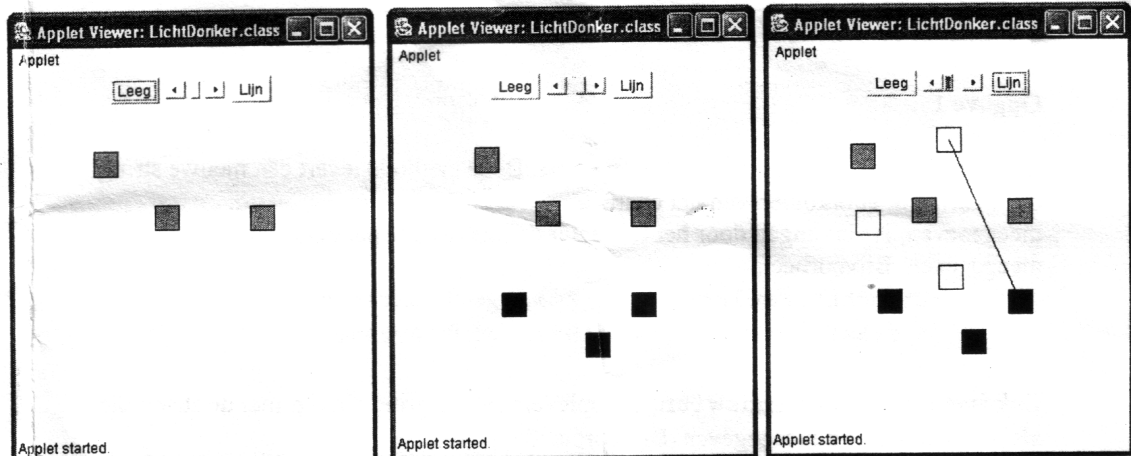
- De HTML-file weglaten: je hoeft alleen de Java-file te schrijven
- De `import`-regels bovenaan het programma weglaten
- De muis-methoden die een lege body hebben weglaten

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 button met het opschrift "Leeg- een schuifregelaar  
- een button met het opschrift "Lijn"
2. Aan het begin staat het schuivertje van de schuifregelaar ongeveer in het midden.
3. Iedere keer als de gebruiker ergens klikt ontstaat er, gecentreert op dat punt, een vierkant met lengte en breedte 20.
4. De kleur van het vierkant kan allerlei grijstinten aannemen, met wit en zwart als uitersten. De grijswaarde wordt bepaald door de stand van de schuifregelaar op het moment dat het punt wordt aangeklikt: hoe verder naar links, hoe lichter. De schuifregelaar helemaal naar rechts is zwart.
5. Rondom de vierkantjes zit altijd een zwarte rand.
6. Er zijn maximaal 100 vierkanten zichtbaar. Als de gebruiker daarna toch meer punten aanklikt, gebeurt er niets (maar er mag ook geen run-time fout optreden!).
7. Na het indrukken van de "Lijn"knop verschijnt er een lijn tussen het lichtste en het donkerste vierkant. Als er twee vierkanten even licht of donker zijn, kies je daarvan het vierkant dat het eerst was aangeklikt.
8. Na het nogmaals indrukken van "Lijn" verdwijnt de lijn weer, bij de derde klik verschijnt hij weer, enz.

9. Na het indrukken van "Leeg" verdwijnen alle vierkanten. De gebruiker kan dan weer met 100 nieuwe kliks beginnen.



Op de afbeelding zie je achtereenvolgens de situatie:

- Na het aanklikken van 3 punten: er verschijnt een grijs vierkant (want de schuifregelaar staat nog in het midden) met een zwarte rand op de aangeklikte plaatsen.
- Na het verschuiven van het schuivertje helemaal naar rechts, en het aanklikken van nog drie punten: hier verschijnen zwarte vierkanten.
- Na het verschuiven van het schuivertje helemaal naar links, het aanklikken van nog drie punten, en het indrukken van "Lijn": er verschijnen witte vierkanten, en er komt een lijn van het eerst aangeklikt witste vierkant naar het eerst aangeklikte zwartste vierkant.