

Inleiding Programmeren

College 13

Universiteit van Amsterdam

Robin de Vries

18 oktober 2017



Inhoud

Hoe je voor te bereiden op het tentamen?

Recursie revisited

Recursie met bomen

Overzicht van het vak

Oefenvraagstukken: oefentoets

Oefenvraagstukken: deoltoets 2 van 2013

Nawoord



Hoe je voor te bereiden op het tentamen?

Stap 1 - voorbereiding:

- Begin op tijd
- Breng in kaart wat de onderwerpen zijn
- Zorg dat je bij bent met de stof (practica af)
- Inventariseer knelpunten: wat snap je nog niet?
- Stel vragen tijdens vragenuurtjes (nu dus!)



Hoe je voor te bereiden op het tentamen? (2)

Stap 2 - uitvoering:

- Sluit jezelf af van je omgeving, zet je telefoon op stil, tv uit
- Bekijk alle slides: snap je wat er gebeurt?
- Bekijk de practica: wat heb je gedaan en geleerd?
- Compileer code, probeer dingen uit!
- Stel jezelf vragen: wat als?
- Oefen oude tentamens (zonder de aanname te doen dat jouw tentamen er vergelijkbaar uitziet)



Hoe je voor te bereiden op het tentamen? (3)

Stap 3 - afronding:

- Schrijf de dingen die je niet kunt onthouden op een spiekbriefje.
- Ga optijd naar bed.
- Vertrek op tijd van huis.
- Neem je ID, voldoende pennen en wat eten/drinken mee.
- Niet te veel stressen ;-)



Inhoud

Hoe je voor te bereiden op het tentamen?

Recursie revisited

Recursie met bomen

Overzicht van het vak

Oefenvraagstukken: oefentoets

Oefenvraagstukken: deelttoets 2 van 2013

Nawoord



Recursie

Bij functies die gebruikmaken van **recursie** roept de functie zichzelf aan.

Laten we eens terugkijken naar de slides van college 6.



Recursieve String-omdraaier

Hoe draai je recursief een String om?

```
static String reverse(String s)
{
    if(s.length() == 0)
        return s;

    return reverse(s.substring(1)) + s.charAt(0);
}
```



Inhoud

Hoe je voor te bereiden op het tentamen?

Recursie revisited

Recursie met bomen

Overzicht van het vak

Oefenvraagstukken: oefentoets

Oefenvraagstukken: deoltoets 2 van 2013

Nawoord



Boom

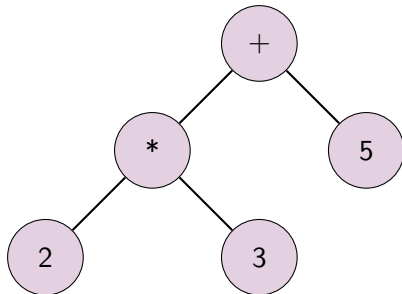
We hebben al gekeken naar **grafen**, iets eenvoudiger zijn bomen. Een boom bestaat uit nodes die kinderen kunnen hebben. Iedere node heeft slechts één ouder.



Boom (2)

We kunnen een boom bijvoorbeeld gebruiken om het rekensommetje “ $2*3+5$ ” te representeren:
Hierbij hebben we drie mogelijke “traversals”:

- prefix: $23*5+$
- infix¹: $2*3+5$
- postfix: $+*235$



¹Dit werkt uiteraard alleen als een node 0 of 2 kinderen heeft, dat noemen we een binaire boom.

Boom in Java

We kunnen vervolgens zo'n boom redelijk eenvoudig representeren:

```
class Node {
    ArrayList<Node> children = new ArrayList<Node>();
    String text;

    Node(String text, Node... children){
        this.text = text;
        this.children.addAll(Arrays.asList(children));
    }
}

// Vervolgens kunnen we ons sommetje eenvoudig maken:
Node tree = new Node("+", new Node("*", new Node("2"), new Node("3")),
                    new Node("5"));
```



Tree traversal

Kunnen we (recursieve) algoritmen schrijven waarmee we de drie verschillende traversals kunnen uitvoeren?



Prefix/Postfix traversal

```
static void traverse_prefix(Node node) {  
    for(Node c : node.children)  
        traverse_prefix(c);  
    System.out.print(node.text);  
}
```

```
static void traverse_postfix(Node node) {  
    System.out.print(node.text);  
    for(Node c : node.children)  
        traverse_postfix(c);  
}
```



Infix traversal

```
static void traverse_infix(Node node) {  
  
    if(node.children.size() != 0){  
        System.out.print("(");  
        traverse_infix(node.children.get(0));  
    }  
  
    System.out.print(node.text);  
  
    if(node.children.size() != 0) {  
        traverse_infix(node.children.get(1));  
        System.out.print(")");  
    }  
}
```



Inhoud

Hoe je voor te bereiden op het tentamen?

Recursie revisited

Recursie met bomen

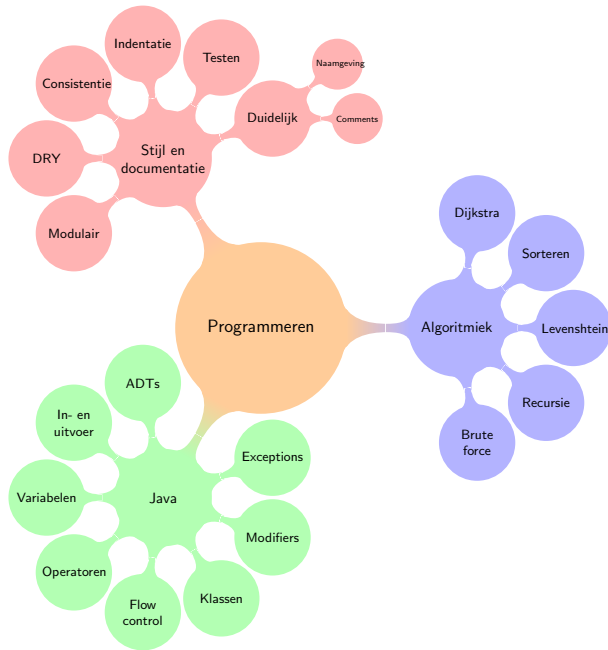
Overzicht van het vak

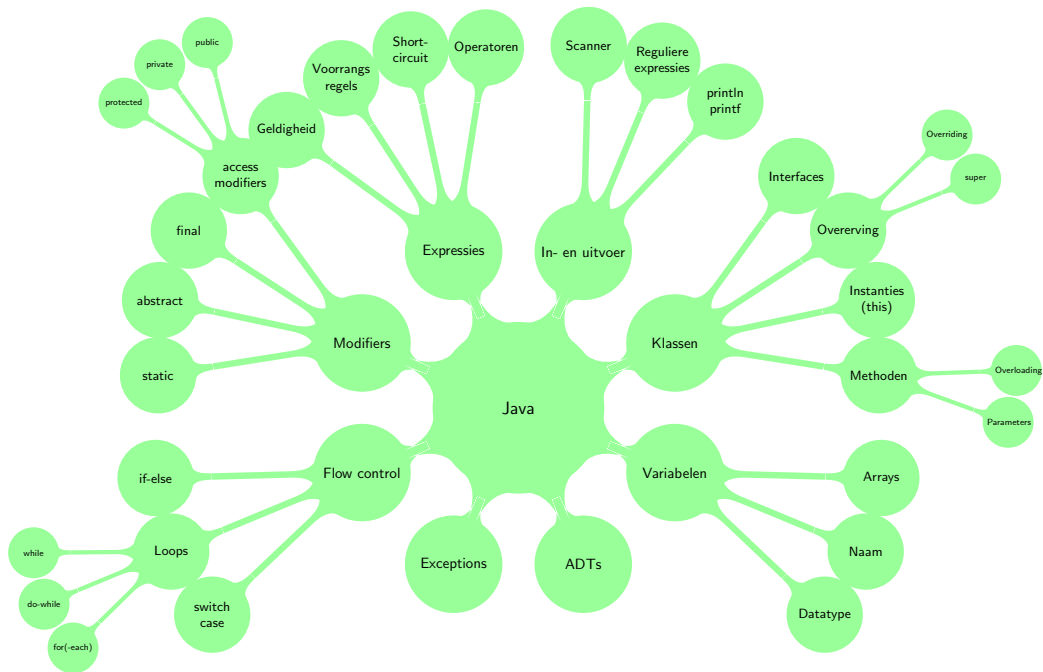
Oefenvraagstukken: oefentoets

Oefenvraagstukken: deelttoets 2 van 2013

Nawoord







Inhoud

Hoe je voor te bereiden op het tentamen?

Recursie revisited

Recursie met bomen

Overzicht van het vak

Oefenvraagstukken: oefentoets

Oefenvraagstukken: deelttoets 2 van 2013

Nawoord



Vraag 1p: Wat is de uitvoer van onderstaande code?

```
class Tester {  
    int var;  
    Tester(double var) {  
        this.var = (int)var;  
    }  
    Tester(int var) {  
        this("hello");  
    }  
    Tester(String s) {  
        this();  
        System.out.print(s);  
    }  
    Tester() {  
        System.out.print("good-bye");  
    }  
    public static void main(String[] args) {  
        Tester t = new Tester(5);  
    }  
}
```



Vraag 1t: Wat is de uitvoer van het volgende programma?

```
public class Opgave {  
    public static void main(String[] args){  
        TelOp st1 = new TelOp();  
        TelOp st2 = new TelOp();  
        TelOp st3 = new TelOp();  
  
        System.out.println(st1.nr + " en " + st1.nrs);  
        System.out.println(st2.nr + " en " + st2.nrs);  
        System.out.println(st3.nr + " en " + st3.nrs);  
    }  
}  
  
class TelOp {  
    int nr;  
    static int nrs;  
  
    TelOp() {  
        nr++;  
        nrs++;  
    }  
}
```



Vraag 1v: Wat is de uitvoer van onderstaande code?

```
class Punt {  
    int x, y;  
    Punt(int x, int y) {  
        this.x = x;  
        this.y = y;  
        System.out.println("4");  
    }  
    Punt(double x, double y) {  
        this((int) x, (int) y);  
        System.out.println("3");  
    }  
    Punt(int x) {  
        this(x + 0.5);  
        System.out.println("2");  
    }  
    Punt(double x) {  
        this((int) x, 0);  
        System.out.println("1");  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        Punt p = new Punt(0);  
    }  
}
```



Vraag 1z: Wat is de uitvoer van onderstaande code?

```
public class Test {  
    public static void main(String [] args) {  
        P p1 = new P(9);  
        P p2 = new P(7);  
        System.out.println(p1.x + " " + p2.x);  
  
        swap(p1, p2);  
        System.out.println(p1.x + " " + p2.x);  
    }  
    static void swap(P p, P q) {  
        p.x = 10;  
        P temp = p;  
        p = q;  
        q = temp;  
    }  
}
```

```
class P {  
    int x;  
    P(int x) {  
        this.x = x;  
    }  
}
```



Inhoud

Hoe je voor te bereiden op het tentamen?

Recursie revisited

Recursie met bomen

Overzicht van het vak

Oefenvraagstukken: oefentoets

Oefenvraagstukken: deelttoets 2 van 2013

Nawoord



Vraag 1: Wat is de uitvoer van onderstaande code?

```
class Parent {  
    void p(double d) {  
        System.out.println(d * 2);  
    }  
    void q(double d) {  
        System.out.println(d * 4);  
    }  
}  
class Child extends Parent {  
    void p(double d) {  
        System.out.println(d);  
    }  
    void q(int i) {  
        System.out.println(i * 3);  
    }  
}
```

```
class Opgave1 {  
    public static void main(String[] args) {  
        Child c = new Child();  
        c.p(10);  
        c.p(10.0);  
        c.q(10);  
        c.q(10.0);  
    }  
}
```



Vraag 2: Welke aanroepen zijn niet toegestaan?

```
class Opgave2 {  
    public static void main(String[] args) {  
        Mammal m = new Dog();  
        Dog d = new Dog();  
        Cat c = new Cat();  
  
        m = d;  
        d = m;  
        d = (Dog)m;  
        c = (Dog)m;  
        m = (Dog)m;  
    }  
}  
  
class Mammal {  
}  
  
class Dog extends Mammal {  
}  
  
class Cat extends Mammal {  
}
```



Vraag 3: Gegeven het volgende programma

```
public class Opgave3 {  
    public static void main(String[] args) {  
        try {  
            int som = Integer.parseInt(args[0]);  
            for (int i = 1; i < args.length; i++) {  
                som = som + Integer.parseInt(args[i]);  
            }  
            System.out.println(som);  
        }  
        catch (Exception ex) {  
            System.out.println("Er gaat iets fout");  
        }  
        System.out.println("Einde");  
    }  
}
```

- 1 Wat haalt dit programma zijn argumenten vandaan?
- 2 Geef een voorbeeld van een goede aanroep van dit programma?
- 3 Wat is in dat geval de uitvoer?
- 4 Geef een voorbeeld van een foute aanroep van dit programma?
- 5 Beschrijf wat nu de uitvoer is.



Vraag 4: Compileert onderstaande programma?

```
class Opgave4 {  
    public static void main(String[] args) {  
        B b = new B();  
        System.out.println(b);  
    }  
}  
  
class A {  
    String s = "Hallo";  
  
    A(String s) {  
        this.s = s;  
    }  
  
    public String toString() {  
        return s + " in A";  
    }  
}  
  
class B extends A {  
    String s = "Hoi";  
}
```



Vraag 5: Bekijk de volgende methode

```
static int methode() throws Exception {  
    int waarde = 0;  
    try {  
        waarde = 1;  
        doIets();  
        waarde = 2;  
    }  
    catch (NullPointerException e) {  
        waarde = waarde + 10;  
    }  
    catch (RuntimeException e) {  
        waarde = waarde + 20;  
    }  
    return waarde;  
}
```

Welke waarde geeft de methode terug als doIets():

- ① een NullPointerException op gooit?
- ② een ArithmeticException op gooit?
- ③ een Exception op gooit?
- ④ met succes wordt uitgevoerd?



Vraag 6: Wat is de uitvoer van onderstaande code?

```
class Krant {
    String nieuws = "Krantennieuws";
    String bericht = "Krantenbericht";
    static String reclame() {
        return "reclame in Krant";
    }
    String getNieuws() {
        return nieuws;
    }
    String getBericht() {
        return bericht;
    }
}
```

```
class Trouw extends Krant {
    String nieuws = "Trouwnieuws";
    String bericht = "Trouwbericht";
    static String reclame() {
        return "reclame in Trouw";
    }
    String getNieuws() {
        return nieuws;
    }
}

Krant k = new Trouw();
System.out.println("1 " + k.nieuws);
System.out.println("2 " + ((Trouw)k).nieuws);
System.out.println("3 " + k.reclame());
System.out.println("4 " + ((Trouw)k).reclame());
System.out.println("5 " + k.getNieuws());
System.out.println("6 " + k.getBericht());
```



Vraag 7: Wat is de uitvoer van de volgende code?

```
class A {  
    int i;  
    static int j;  
    A() {  
        i = 2;  
        j = 1;  
    }  
    void m() {  
        i = 4;  
    }  
    static void m1() {  
        j = 6;  
    }  
}  
A a = new B();  
System.out.println(a.i);  
System.out.println(((B)a).i);  
System.out.println(A.j);  
System.out.println(B.j);
```

```
class B extends A {  
    int i;  
    static int j;  
    B() {  
        i = 3;  
        j = 5;  
    }  
    void m() {  
        i = 7;  
    }  
    static void m1() {  
        j = 9;  
    }  
}  
a.m();  
a.m1();  
System.out.println(a.i);  
System.out.println(((B)a).i);  
System.out.println(A.j);  
System.out.println(B.j);
```



Vraag 8: Wat is de uitvoer van de volgende code?

```
class Parent {
    public String toString() {
        return("Parent");
    }
}

class Child extends Parent {
    public String toString() {
        return(super.toString() + " en Child");
    }
}

class GrandChild extends Child {
    public String toString() {
        return(super.toString() + " en GrandChild");
    }
}
```

```
Parent p = new Parent();
Child c = new Child();
GrandChild g = new GrandChild();
```

```
Parent q = c;
System.out.println(q);
Child r = g;
System.out.println(r);
```



Vraag 9: Welke van de toekenningen zijn niet toegestaan?

```
interface Verplaatsbaar {  
    public void verplaats(int d);  
}  
  
class Tafel implements Verplaatsbaar {  
    public void verplaats(int ri) {  
  
    }  
}  
  
class Stoel implements Verplaatsbaar {  
    public void verplaats(int ri) {  
  
    }  
}
```

```
Verplaatsbaar v1, v2;  
Tafel ta = new Tafel();  
Stoel st = new Stoel();
```

```
v1 = (Verplaatsbaar)ta;  
v2 = (Verplaatsbaar)st;  
ta = (Stoel)v1;  
st = (Stoel)v2;
```



Inhoud

Hoe je voor te bereiden op het tentamen?

Recursie revisited

Recursie met bomen

Overzicht van het vak

Oefenvraagstukken: oefentoets

Oefenvraagstukken: deelttoets 2 van 2013

Nawoord



Nawoord

Heel veel succes met het tentamen!

