

HerhalingsLes



| | |
|-----------|--|
| Klasgroep | 1EO-ICT |
| Opleiding | Bachelor Elektronica-ICT |
| Lokaal | Groot Auditorium |
| Tijdstip | maandag lestijd 3 |
| Docent | Katja Verbeeck |
| Contact | katja.verbeeck@odisee.be |
| Handboek | hfst 1 → 6 (alleen de onderwerpen die op de slides vernoemd worden - bvb geen recursie of bitoperatoren) |

Inhoud

- 1 Introductie
- 2 Java Fundamentals
- 3 Data Types en Operaties
- 4 Controle Structuren
- 5 Arrays
- 6 Klassen, methoden & objecten
- 7 Vervolg methoden
- 8 Oefeningen



Java Fundamentals : inleiding

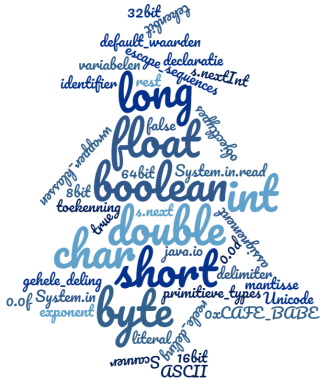


Java Fundamentals : inleiding

Verzin zelf je vragen !

- Hoe compileer je een java file ?
- Hoe run je een java programma ?
- Wat zijn de stijlfaspraken?
- Welke verschillende soorten fouten kunnen er optreden en wanneer ?
- ...

Primitieve Types en operatoren in Java



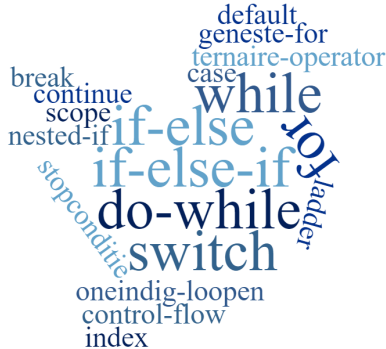
Primitieve Types en operatoren in Java

Verzin zelf je vragen !

- Wat zijn de primitieve types? Welke waarden kunnen ze bijhouden? Hoeveel geheugen nemen ze in beslag?
- Wat zijn object of referentie types?
- Wat zijn logische operatoren?
- Wat zijn relationele operatoren?
- Wat is het verschil tussen een gehele en reële deling?
- ...

van 2

Selecties en herhalingen



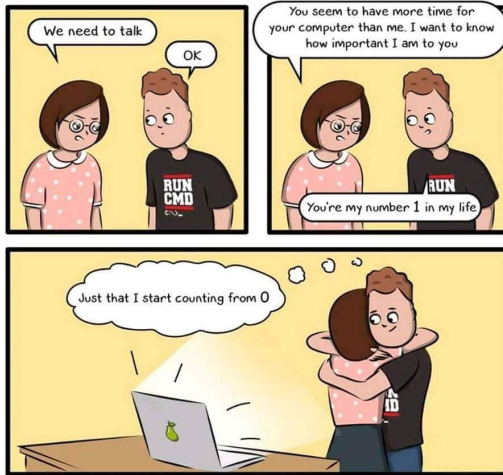
Bepaalde blokken code wil je herhalen, andere blokken wil je voorwaardelijk maken (alleen uitvoeren als aan een bepaalde conditie voldaan is).

Arrays



Deze **datastructuur** laat toe om via 1 variabele een hele rij van homogene data (= data van hetzelfde type) te manipuleren

Onthoud : tellen vanaf 0 !



fb.com/programmingjokes

NERD4LIFE.studio

Klassen, methoden & objecten



Programma's bestaan uit meer dan 1 klasse - een demo klasse met main methode + enkele andere nieuwe types. Een nieuw type (class) bestaat uit data + methoden om die data te manipuleren. Constructoren zijn byzondere methoden om objecten te maken.

ethode-
self, or

oef: de klasse WoordSpel (labo 11)

Schrijf een programma dat een gebruiker moet laten raden naar een woord. Het te raden woord wordt willekeurig gekozen uit een array die vooraf gevuld werd met mogelijke woorden. De gebruiker geeft ofwel een letter in ofwel het volledige woord wanneer hij denkt het woord te kennen. Voorzie een constante om het maximaal aantal beurten vast te leggen. Zolang het woord niet geraden werd en het maximum aantal beurten niet overschreden werd mag de speler verder raden.

Datavelden

- een constante met het maximum aantal beurten
- een constante array opgevuld met 10 verschillende woorden
- het huidige woord dat geraden moet worden
- een booleaanse array die bijhoudt welke letter uit het huidige woord al geraden werd.
- het aantal gokbeurten

Voorzie alleen wat gevraagd is !

Datavelden

```
public final int MAX;  
private final String[] WOORDEN;  
private String teRadenWoord;  
private boolean[] geradenLetters;
```

Methoden

- een constructormethode om je datavelden te initialiseren.
- een *isGeraden* methode die test of alle booleans van de booleaanse array op true staan. Deze methode neemt geen parameters en geeft zelf ook een booleaanse waarde terug.
- een *gok* methode die test of de meegegeven letter in het woord voorkomt. Indien ja, dan moet de booleaanse array aangepast worden. De methode geeft een String terug die alle tot dan toe correct geraden letters weergeeft. De andere letters worden verstopt achter een underscore karakter.
- een *gok* methode die test of het meegegeven woord het te raden woord is. Deze versie van de gok methode is een **overload versie** van de vorige. Deze methode neemt nu een string als parameter ipv een karakter. Als output geeft deze methode opnieuw een String terug. Wanneer de gok correct was, wordt het te raden woord volledig weergegeven en de booleaanse array wordt aangepast. Indien de gok niet juist was, wordt opnieuw de string met alle tot dan toe geraden karakters weergegeven.

Constructoren

```
public WoordSpel() {
    MAX = 5;
    WOORDEN = new String[]{"paddenstoel" , "breedsmoelkikker" , "paardenmelk" ,
        "hoefsmid" , "huisjesmelker",
        "aardbei", "zeepaardje", "sinterklaas", "tovenaar",
        "armoedebestrijding"};
    teRadenWoord = WOORDEN[(int)(Math.random() * WOORDEN.length)];
    geradenLetters = new boolean[teRadenWoord.length()];
}

public WoordSpel(String[] woorden){
    MAX = 5;
    WOORDEN = woorden;
    teRadenWoord = WOORDEN[(int)(Math.random() * WOORDEN.length)];
    geradenLetters = new boolean[teRadenWoord.length()];
}
```

isGeraden

```
public boolean isGeraden(){  
    for(boolean letter : geradenLetters){  
        if (letter) continue;  
        else return false;  
    }  
    return true;  
}
```

Gok versie 1

```
public String gok(char letter) {  
    for (int i = 0; i < teRadenWoord.length(); i++) {  
        if (teRadenWoord.charAt(i) == letter) {  
            geradenLetters[i] = true;  
        }  
    }  
    return maskeerWoord();  
}  
  
private String maskeerWoord(){  
    String antwoord = "";  
    for (int i = 0; i < geradenLetters.length; i++) {  
        antwoord += geradenLetters[i]?teRadenWoord.charAt(i) : "-";  
    }  
    return antwoord;  
}
```

Gok versie 2

```
public String gok(String woord) {  
    if (teRadenWoord.equals(woord)) {  
        for (int i = 0; i < teRadenWoord.length(); i++)  
        {  
            geradenLetters[i] = true;  
        }  
        return teRadenWoord;  
    }  
    else return maskeerWoord() ;  
}
```

Demo van het spel

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    WoordSpel spel = new WoordSpel();
    String antwoord;
    int aantalBeurten = 0;

    do {
        aantalBeurten++;
        System.out.print("Geef je letter of het volledige woord: ");
        antwoord = sc.nextLine();
        if (antwoord.length() == 1) {
            antwoord = spel.gok(antwoord.charAt(0));
        } else antwoord = spel.gok(antwoord);
        if (spel.isGeraden()) {
            System.out.println("Proficiat!\nJe vond het woord in " + aantalBeurten +
                               " trials");
            return;
        }
        System.out.println("Beurt " + aantalBeurten + ": " + antwoord);
    } while (aantalBeurten <= spel.MAX);

    System.out.println("Helaas!\nJe vond het woord niet in " + spel.MAX + " trials");
}
```

oef : De klasse ICTdocenten

Schrijf een klasse om de namen en voornamen van (ICT) docenten bij te houden. Deze gegevens zal je in je klasse in een dataveld *docenten* bewaren als een dubbele array van String objecten. De input van deze gegevens krijg je via de parameter van de constructor van deze klasse. Voorzie tevens een methode *geefInitialen* om de initialen van alle docenten terug te geven als een array van Strings.

oef : demo ICTdocenten

In je demo programma vraag je de gebruiker een vooraf bepaald aantal keer een voor- en achternaam van een docent op. Gebruik deze info om een object van de Klasse ICTdocenten aan te maken. Stuur dit object het bericht *geefInitialen* en druk deze netjes af op het scherm.

Organiseer je data

Je kan elke rij voorzien om 1 docent te bewaren. In de eerste kolom hou je de voornaam bij in de tweede kolom hou je de achternaam bij.

Bijvoorbeeld :

| | |
|----------|------------|
| Kristien | Van Assche |
| Bram | Gosseye |
| Tim | Vermeulen |

```
//1) declaratie  
String[][] docenten;  
  
//2) instantiatie  
docenten = new String[3][2];
```


Organiseer je data

```
//3) initialisatie van de rij
```

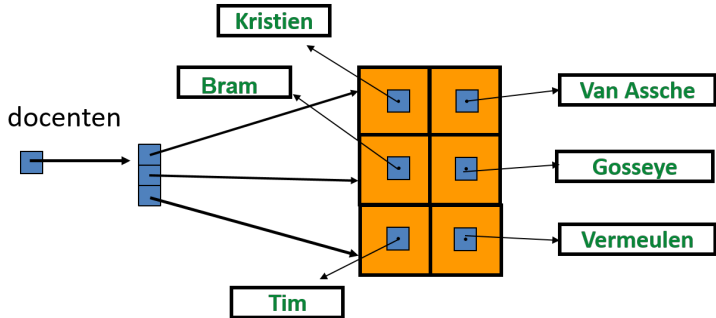
```
docenten[0][0] = "Kristien";  
docenten[0][1] = "Van Assche";  
  
docenten[1][0] = "Bram";  
docenten[1][1] = "Gosseye";  
  
docenten[2][0] = "Tim";  
docenten[2][1] = "Vermeulen";
```

```
//3) of via de initializer syntax:
```

```
String[][] docenten = {  
    {"Kristien", "Van Assche" },  
    {"Bram",      "Gosseye" },  
    {"Tim",       "Vermeulen" } };
```

Vermits het aantal vooraf niet gekend zal zijn, zal er in het demo-programma een for lus gebruikt worden om de namen in te lezen !

Hoe zit dit in het geheugen?



Het demo programma : main methode

```
public static void main(String[] args) {  
  
    Scanner scan = new Scanner(System.in);  
    System.out.println("Hoeveel docentennamen wil  
        je ingeven ?");  
    int aantal = scan.nextInt();  
    String[][] namen = geefNamenIn(aantal);  
  
    ICTdocenten ict = new ICTdocenten(namen);  
    String[] initialen = ict.geefInitialen();  
    for(int i = 0; i < aantal; i++){  
        System.out.println(initialen[i]);  
    }  
}
```

Het demo programma : geefNamenIn methode

```
public static String[][] geefNamenIn(int aantal)
{
    String[][] docenten = new String[aantal][2];
    Scanner scan = new Scanner(System.in);
    for (int i = 0; i < docenten.length; i++) {
        System.out.print("Voornaam: ");
        docenten[i][0] = scan.nextLine();
        System.out.print("Familiennaam: ");
        docenten[i][1] = scan.nextLine();
    }
    return docenten;
}
```

ICTdocenten klasse

```
public class ICTdocenten {  
    String[][] docenten;  
  
    public ICTdocenten (String[][] namen){  
        docenten = namen;  
    }  
  
    ...  
}
```

ICTdocenten klasse

```
...  
  
public String[] geefInitialen(){  
    String [] initialen = null;  
    if(! (docenten == null)){  
        initialen = new String[docenten.length];  
        for (int i = 0; i < docenten.length; i++) {  
            initialen[i] = "" + docenten[i][0].charAt(0);  
            String famNaam = docenten[i][1];  
            int idx;  
            do{  
                initialen[i] += famNaam.charAt(0);  
                idx = famNaam.indexOf(' ');  
                famNaam = famNaam.substring(idx + 1);  
            } while(idx != -1);  
        }  
    }  
    return initialen;  
}
```

Theorie-Examen

- toepassing van de geziene theorie uit de lessen
- slechts zeer beperkte codefragmenten zelf schrijven
- gericht op inzicht en correct gebruik van de geziene technieken
- syntax of logische fouten zoeken in code
- basiskennis van courante syntax verwacht
- geen hulpmiddelen
- duurt max 1 uur

Individueel Labo

zelfde systeem als de individuele oefeningen. Jullie krijgen hier max 2u tijd.

Veel succes met de examens en natuurlijk ...

Prettige feestdagen !!!