

Software Ontwikkeling Labo 1:

Value Types & Controlestructuren

Methods & Arrays

Algemene afspraken labo's

- a) op het einde van het labo worden de uitgewerkte oplossingen op de dropbox van minerva geplaatst.
- b) de niet uitgewerkte oplossingen worden zelfstandig uitgewerkt tegen het volgende labo en ingediend op het einde van het volgende labo (tenzij anders vermeld wordt).
- c) noteer ook telkens de tijd die je aan elke oefening gewerkt hebt in de code van de oplossing. Dit zal voor een klein deel bijdragen aan de finale punten.

Oefeningen

1. Hello

Schrijf een programma dat uitschrijft: "Hallo vreemdeling"

Dat daarna (volgende lijn) uitschrijft: "Hoe heet je?"

Dat daarna de input van de gebruiker opslaat en tenslotte uitschrijft: "Hallo *NAAM*"

2. Inhoud

Schrijf een programma dat de inhoud van een balk berekent aan de hand van de drie afmetingen die daarvoor aan de gebruiker gevraagd worden.

Doe daarna hetzelfde voor een cilinder. Je kan hiervoor de waarde van de constante `PI` terugvinden onder `Math.PI`, die je gewoon als een andere variabele kan gebruiken. De klasse `Math` heeft overigens ook een aantal statische functies waar je in de toekomst handig gebruik van kunt maken.

3. Hexadecimaal en Binair

Schrijf een programma dat een ingegeven getal in hexadecimale en binaire vorm uitschrijft (versie 1 in `main`)

Schrijf een methode die in staat is om een gegeven getal in een X-delig talstelsel uit te schrijven.

4. Maximum van een reeks

Schrijf een programma dat continue een waarde inleest en die vergelijkt met de grootste reeds ingegeven waarde (de standaard grootste waarde = 0). Print achter elke ingave de grootste waarde op het scherm.

- a) Stop wanneer de ingegeven waarde niet groter is dan de grootste reeds ingegeven waarde.
- b) Stop wanneer de ingegeven waarde kleiner is dan 0.

5. Faculteit

Schrijf een programma dat een getal aan de gebruiker vraagt. Bereken de faculteit van dit getal.

Ter herinnering: $5! = 1*2*3*4*5 = 120$

Schrijf dit in twee verschillende versies:

- a) niet recursief
- b) recursief

6. Meetkundige rij

Een meetkundige rij is in de wiskunde een rij getallen waarin elk volgend element ontstaat door zijn voorganger met een constante (de reden) te vermenigvuldigen. Als a het eerste element is van de rij en r de reden, dan ligt de gehele rij vast.

Maak een programma dat het eerste element, de reden en het aantal termen vraagt aan de gebruiker en op basis daarvan de eerste termen (opgegeven aantal) van de meetkundige rij op het scherm afprint.

Een gebruiker die verkeerde informatie aan dit programma geeft, kan ervoor zorgen dat de berekeningen onbetrouwbaar worden. Waaraan zal je dit merken en leg uit waardoor dit veroorzaakt wordt. Zorg in de code dat de gebruiker verwittigd wordt op het moment dat dit optreedt. Zorg er ook voor dat de code op dit moment de uitvoering staakt.

7. Huizen

In een straat staan 20 huizen. Sommige huizen zijn bewoond, andere niet. De bewoners hebben huurcontracten afgesloten voor 1 jaar, en op het einde van het jaar wordt er heel wat verhuisd. Dit gebeurt volgens het volgende patroon:

- Wanneer iemand het hele jaar twee burens heeft gehad, dan verlaat hij de straat omwille van geluidsoverlast. Heeft hij het vorige jaar geen burens gehad, dan verdwijnt hij uit eenzaamheid. Het huis blijft dus enkel bewoond wanneer precies 1 van de aanpalende huizen bewoond was.
- In een leegstaand huis komt het volgende jaar iemand wonen wanneer er minstens 1 van de naburige huizen het vorige jaar bewoond was. Anders blijft het huis leeg.

Schrijf een programma dat de bewoning van de huizen simuleert over een bepaalde tijdspanne (bijv. 24 jaar). Als invoer neemt het programma 20 cijfers 0 of 1 die aangeven of de resp. huizen bewoond zijn (1) of (0). Daarna wordt elk jaar de situatie schematisch weergegeven (1 lijn per jaar). Een leegstaand huis wordt voorgesteld door twee blanco's, een bewoond huis door "XX ", zoals in het volgende voorbeeld:

```

1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 1
XX          XXXXXXXX          XX
  XX      XXXX  XXXX          XX
XX  XX  XXXXXXXXXXXXXXXXXXXX  XX  XX
  XX  XX  XXXX          XXXX  XX  XX
XX  XX  XXXXXXXX          XXXXXXXX  XX  XX
  XX  XXXX  XXXX  XXXX  XXXX  XX
XX  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  XX

```

Breng de nodige structuur in je programma aan door aparte methoden te voorzien voor:

- de overgang van het ene jaar naar het volgende
- het uitprinten van de huidige situatie

Opmerking: Gebruik voor de overgangen twee tabellen om de straat voor te stellen. Één tabel voor de situatie van vorig jaar en één voor de huidige situatie. Het is makkelijker om tabellen van 22 elementen te gebruiken (genummerd 0 ... 21) om randgevallen op te vangen. Vermijd het gebruik van overdreven veel vernestelde 'if's.

8. Zeef van Eratosthenes

De volgende methode om alle priemgetallen te bepalen tussen 2 en een gegeven bovengrens (bijv. 1000) was al bekend bij de Oude Grieken en wordt nu over het algemeen aangeduid met de benaming 'Zeef van Eratosthenes'.

Beschouw alle getallen tussen 2 en 1000. Schrap alle veelvouden van 2. Het eerste getal dat nu nog niet geschrapt is, is 3. Schrap nu alle veelvouden van 3. Het eerste getal dat nu nog niet geschrapt is, is 5. Schrap nu alle veelvouden van 5. We hebben nu reeds 3 priemgetallen gevonden (2, 3 en 5). Door deze methode verder te volgen, vinden we uiteindelijk alle priemgetallen tussen 0 en 1000.

Schrijf een programma dat op deze manier alle priemgetallen zoekt tussen 0 en 1000 en deze uitschrijft op het scherm.

Schrijf de code zodanig dat het effectieve zoeken naar priemgetallen gebeurt in een aparte methode die een array teruggeeft waar enkel priemgetallen in zitten (dus geen array met zowel priem- als andere getallen).

9. Sorteren

Een van de meest voorkomende operaties in computerprogramma's is het sorteren van een reeks getallen. Voor deze taak zijn dan ook al een heleboel algoritmen bedacht. Bedenk **zelf** (dus zonder eerst te gaan zoeken naar bestaande algoritmen) een manier om een array van gehele getallen te sorteren. Werk je methode eerst uit op papier en implementeer deze daarna.