

Inleiding tot programmeren

 tuyaux.winak.be/index.php/Inleiding_tot_programmeren

Inleiding tot programmeren

Richting	<u>Wiskunde</u>
----------	-----------------

Jaar	<u>1BWIS</u>
------	--------------

Bespreking

Programmeren is in de 1ba wiskunde een vak ik het 2de semester. Men leert er werken met de programmeertaal C++. Je krijgt een basis van de taal mee en bent op het einde van de cursus (normaal gezien) in staat om eender welke code in C++ te lezen (maar daarom niet alles te begrijpen).

Theorie

De theorie wordt in hoorcolleges gegeven door prof. Famaey. Hij geeft les aan de hand van presentaties, die ook ter beschikking gesteld worden op Blackboard. Er is ook een handboek ter beschikking op de cursusdienst. Ik raad niet aan om dit te kopen, aangezien je waarschijnlijk alleen de presentaties zal studeren voor het examen (dit is ook zeker voldoende).

Oefeningen

De practica worden gegeven door Tom Hofkens in samenwerking met Brent van Bladel. Je leert er de code te gebruiken. De practica behandelen normaal de vorige theorieles. Wanneer je niet alle oefeningen in de les afkrijgt, wordt verwacht dat je deze thuis afmaakt. Je zal soms verder moeten gaan op oefeningen die in de vorige les gegeven zijn, daarom is het vrij belangrijk om de oefeningen af te proberen krijgen.

Puntenverdeling en examen

50% van je eindcijfer staat op het project dat je moet maken. Je krijgt de opgave waarschijnlijk vlak voor de paasvakantie. Probeer er zo vroeg mogelijk aan te beginnen, want het is geen gemakkelijke opdracht. Laat je niet meteen afschrikken, want het kan heel moeilijk lijken op het eerste zicht. Probeer het stap voor stap aan te pakken, en op tijd hulp te vragen als je vast zit. De overige 50% van je eindcijfer wordt bepaald door je examen. Dit bestaat voor de helft uit theorievragen en voor de andere helft uit oefeningen. Professor Famaey zal de laatste les nog een voorbeeldexamen tonen en toelichten, daardoor weet je heel goed wat je te wachten staat. Om te slagen voor dit vak moet je voor zowel het project als het examen minstens 8/20 halen (en in het totaal geslaagd zijn natuurlijk).

Vragen

Academiejaar 2018 - 2019

Computer

[Media:inleiding_tot_programmeren-18191.pdf](#)

Academiejaar 2016 - 2017

Computer

1. Schrijf een programma dat wiskundige rijen en reeksen kan voorstellen. We gaan uit van een wiskundige rij van reële getallen die een recursief voorschrift heeft, waarbij elke term afhankelijk is van zijn voorganger. Een voorbeeld is de rekenkundige rij $u = [1, 2, 3, 4, 5, 6, \dots]$, met het voorschrift $u[n] = u[n-1] + 1$. Een reeks is gebaseerd op een rij, waarbij de n -de term van de reeks de som is van de eerste n termen van de rij. De reeks r horende bij de rekenkundige rij u is dan ook $r = [1, 3, 6, 10, 15, 21, \dots]$. Schrijf een programma dat de volgende concepten kan voorstellen:

Een concept rij dat volgende functionaliteiten voorziet:

1. Het opvragen van de huidige term van die rij, die initieel wordt ingesteld op de eerste waarde van de rij (bv. in het voorbeeld wordt de huidige term initieel ingesteld op 1). De initiële waarde van de huidige term moet kunnen ingesteld worden door de gebruiker.
2. Het opvragen van de volgende term van de rij, zonder de huidige term te overschrijven (bv. in het voorbeeld wordt 5 teruggegeven wanneer de huidige term 4 is).
3. Een implementatie van de ++operator die de volgende term van de rij berekent, teruggeeft en opslaat als nieuwe huidige term.
4. Een implementatie van de --operator die de vorige term van de rij berekent, teruggeeft en opslaat als nieuwe huidige term.
5. Het uitschrijven van de volgende n termen (inclusief de huidige term h) van de rij u naar het scherm in de vorm "[$u[h]$, $u[h+1]$, ..., $u[h + n - 1]$]" Na het uitvoeren van de functie moet de huidige term ingesteld staan op de eerstvolgende term die niet is uitgeschreven naar het scherm.
6. Het uitschrijven van de volgende n termen (inclusief de huidige term) van de rij naar een bestand (elk element op een afzonderlijke lijn). Als argumenten worden het aantal uit te schrijven termen n en de naam van het bestand meegegeven. Na het uitvoeren van de functie moet de huidige term ingesteld staan op de eerstvolgende term die niet is uitgeschreven naar het bestand.

Een rekenkundige rij is een speciale rij waarbij de volgende term berekend wordt door er een constant verschil v bij op te tellen. $u[n] = u[n-1] + v$.

1. Het type van het verschil v (i.e., geheel getal of reëel getal) moeten kunnen worden opgegeven door gebruik te maken van templates.
 - Een meetkundige rij is een speciale rij waarbij de volgende term berekend wordt door er een constante reden r mee te vermenigvuldigen. $u[n] = u[n-1] \times r$.
 - De harmonische rij is een speciale rij: $[1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots]$ waar $u[n] = \frac{1}{n}$.
 - Een reeks kan worden voorgesteld als een soort rij. Om de volgende term te berekenen, gebruikt die een rij: de volgende term van de reeks is de huidige term van de reeks + de volgende van de bijbehorende rij.
 - Een functie voor het aanmaken van een rij op basis van informatie die wordt ingelezen uit een bestand. De naam van het bestand wordt meegegeven als parameter. Je mag zelf beslissen hoe de informatie in het bestand wordt voorgesteld, maar voeg minstens 1 voorbeeldbestand toe aan je ingediende oplossing. Dit bestand moet volgende informatie bevatten:
 1. De waarde van de eerste term van de rij
 2. Het type van de rij (i.e., rekenkundig, meetkundig, harmonisch)
 3. Of het een rij of een reeks is

Voorzie daarenboven een `main()` functie waarin je de werking demonstreert van elk van de gevraagde functionaliteiten. Maak gebruik van een goede programmeerstijl, en deel je code op een juiste manier op in verschillende bestanden (inclusief header files). Vertaal de concepten in een geschikte klassenstructuur en maak gebruik van overerving waar nuttig.

Academiejaar 2015-2016

Computer

1. Schrijf een programma dat de rij van Fibonacci genereert. Een rij van Fibonacci bestaat uit een reeks positieve gehele getallen waarbij elk daaropvolgend getal de som is van de twee voorgaande. Het programma moet de volgende functionaliteiten bevatten:
 - Een functie `vector<int> fibonacci(int max)` die een vector teruggeeft met de Fibonacci rij tot een opgegeven maximale waarde. De variabele `max` stelt niet het aantal getallen in de rij voor, maar de maximale waarde van het grootste getal.
 - Een `main` functie die de waarde `max` inleest van standard input, een rij van Fibonacci genereert tot deze maximale waarde en deze tenslotte uitschrijft naar het scherm en een bestand met naam "fibonacci_aantal.txt", waarbij aantal wordt vervangen door het aantal getallen in de gegenereerde rij (bv. gegeven de Fibonacci rij "0, 1, 1, 2, 3, 5", zal de naam van het bestand "fibonacci_6.txt" zijn).

2. Schrijf een programma om bibliotheken te beheren

- Dit programma bevat de volgende concepten
 - Een bibliotheek bestaande uit een lijst van boeken en een lijst van klanten
 - Een klant wordt geïdentificeerd puur op basis van een klantnummer. Er moet geen andere informatie over de klant worden bijgehouden.
 - Elk boek heeft de volgende eigenschappen: titel (je mag aannemen dat deze uniek is), auteur, genre (Drama, Thriller, Romantisch, SciFi of Fantasy), status (Uitgeleend of Aanwezig), uitlener (het klantnummer van de persoon die het boek op dit moment in zijn bezit heeft).
- Het programma moet daarbovenop de volgende functionaliteiten voorzien:
 - Toevoegen van nieuwe boeken aan de bibliotheek
 - Verwijderen van boeken op basis van hun titel
 - Overschrijf de == operator voor boeken, ze zijn gelijk bij gelijke titel
 - Auteur, genre, status en uitlener van boeken opvragen op basis van hun titel
 - Lijst van uitgeleende boeken opvragen van een persoon op basis van klantnummer
 - Een boek uitlenen aan een persoon op basis van klantnummer en boektitel
 - Een uitgeleend boek terugbrengen op basis van klantnummer en boektitel
- Voorzie daarenboven een main functie die een bibliotheek met een lijst van boeken en klanten aanmaakt (minimaal vijf van elk). Demonstreer hier ook de werking van elk van de gevraagde functionaliteiten.
- Maak gebruik van een goede programmeerstijl en vertaal de concepten in een geschikte klassenstructuur.

Academiejaar 2014 - 2015 1ste zit

Schriftelijk deel

1. Wat doet deze code. Leg uit

```
inti=0;intj=5;while((i++*j--)>=0)if((i*j)!=0)cout<<Math.power(i,j)<<" ";
inti=0;intj=5;while((i++*j--)>=0)if((i*j)!=0)cout<<Math.power(i,j)<<" ";
```

2. Wat is de output van deze code?

```
inta[15];a[-2]=100;a[6]=254;int*p=&a[5];*p=7;p[1]=666;p[-4]=15;p+=3;p[-4]=3;for(inti=0;i<=15;i++)cout<<a[i++-2]<<" ";
inta[15];a[-2]=100;a[6]=254;int*p=&a[5];*p=7;p[1]=666;p[-4]=15;p+=3;p[-4]=3;for(inti=0;i<=15;i++)cout<<a[i++-2]<<" ";
```

3. Is dit een voorbeeld van een goede code? Leg uit.

4. Bespreek wat deze code doet.

```
int telop (vector<int>&r,int x,int y){r[x]+=r[y];return x;}
int telop (vector<int>&r,int x,int y){r[x]+=r[y];return x;}
```

```
vector<int>maak_reeks(int x){vector<int>reeks(x);reeks[0]=1;reeks[1]=1;for(int i=2;i<x;i++){reeks[i]=telop(reeks,i-1,i-2);}return reeks;}
vector<int>maak_reeks(int x){vector<int>reeks(x);reeks[0]=1;reeks[1]=1;for(int i=2;i<x;i++){reeks[i]=telop(reeks,i-1,i-2);}return reeks;}
int main(){vector<int>reeks=maak_reeks(10);for(int i=0;i<10;i++)cout<<reeks[i]<<" ";}
int main()
{vector<int>reeks=maak_reeks(10);for(int i=0;i<10;i++)cout<<reeks[i]<<" ";
```

Computer

1. Schrijf een programma met devolgende functionaliteit:

- Een gebruiker kan lijn per lijn tekst ingeven. Het programma stopt wanneer de gebruiker het woord STOP of stop ingeeft.
- Telkens een nieuwe lijn werd ingetypt (en afgesloten met een enter), wordt de ingetypte tekst omgezet als volgt: hoofdletters worden omgezet in kleine letters en kleine letters worden omgezet in hoofdletters. Het resultaat moet afgeprint worden op het scherm. TIP: Er zijn verschillende manieren om dit te doen. Een ervan is door gebruik te maken van de ASCII tabel van karakters.

2. Schrijf een programma dat op een object-georiënteerde manier een drankautomaat nabootst. De concepten zijn devolgende:

- Een consument heeft een bepaald bedrag in zijn portefeuille.
- Elke type drankje heeft een bepaalde prijs.
- Elke automaat heeft
 - 1. Een reeks van drankjes van verschillende types.
 - 2. Een hoeveelheid geld in de automaat zitten.

Binnen dit programma moet het mogelijk zijn om

1.
 1. Een overzicht te krijgen van de bestaande drankjes met hun prijs en hoeveelheid.
 2. Een drankje te bestellen. Uiteraard kan dit enkel als de gebruiker voldoende geld heeft. Het effect van een drankje te bestellen is dat
 3. De gebruiker betaalt aan de automaat en dus minder geld tot zijn beschikking heeft.
 4. De automaat een "drankje" uitwerpt.
2. Schrijf een functie `resize` met volgende argumenten:
 - Een pointer naar een geheugenplaats van gehele getallen.
 - De grootte van de bestaande geheugenplaats.
 - De grootte van de nieuwe geheugenplaats.

De `resize` functie moet een nieuw blok geheugen aanmaken met een nieuwe grootte en alle bestaande elementen kopiëren naar de nieuwe geheugenplaats. De pointer naar de nieuwe geheugenplaats van gehele getallen is het resultaat van deze functie.

Academiejaar 2012 - 2013 1ste zit

1. Een type heeft 3 betekenissen. Welke? Geef een voorbeeld.
2. Welke conversies zijn veilig? Welke onveilig? Geef van elk 3 voorbeelden.
3. Wat is een compound statement? Waar komt die zoal voor? Wees volledig!
4. Waarvoor dient in de 'switch instructie' de break instructie? Wat gebeurt er wanneer men deze niet gebruikt?
5. Hoe kan men de onderdelen van een vector oproepen? Wat zijn de twee manieren? Is er een beste, en zo ja waarom?
6. Waarom wordt een header file zowel in de implementatie code als in de user code geïnccluded?
7. Welke soorten formele parameters zijn er in C++ voor functies? Wat zijn de vereisten voor de actuele parameter?
8. Waarvoor kan men de using directive gebruiken? Kan er na gebruik ervan nog een name clash optreden?
9. Waarom zou men memberdata van een klasse private maken? Waarom public?
10. Kan men meerdere constructoren hebben in een klasse? Zo ja, wat onderscheidt ze?
11. Waarvoor dienen iostreams? Welk van de twee is het meest 'foutgevoelig' en waarom?
12. Leg het grafische systeem uit. Hoe wordt er dus een object getekent nadat de Display engine geactiveerd is? Wat zou een gebruiker die een nieuwe Shape maakt eventueel moeten doen?
13. Waar moet men bij het werken met pointers en arrays op letten bij toegang tot de individuele onderdelen ervan?
14. ... (uitleg) Hoe kan men memory leaks vermijden?
15. Wat is een copy assignment? Waar dient deze essentieel voor? Hoe kan men deze aanmaken in een klasse?

Academiejaar 2013 - 2014 1ste zit

Elke vraag dient met een uitleg van ± een half blad beantwoord te worden. De laatste vraag (essayvraag) dient heel grondig behandeld te worden (lees: uitleg van ± 2 bladzijden).

1. In c++ zijn conversies mogelijk onder bepaalde voorwaarden. Bespreek en geef voorbeelden.
2. Wat is het verschil tussen een declaratie en een definitie.
3. Waarom wordt een header-file zowel in de user code als in de implementatie-file gebruikt?
4. Bespreek de verschillende formele parameters. Wat zijn de voorwaarden voor de actuele parameters?
5. Wat is een namespace? Waar wordt deze voor gebruikt en wat zijn de voordelen? Leg ook uit hoe je onderdelen uit een namespace kunt kwalificeren.
6. Bespreek waarom 'class Shape' een abstracte klasse is. Hoe kan je een abstracte klasse maken?
7. Wat is een 'copy-constructor'? Waar dient deze essentieel voor? Hoe maak je er een (geen exacte code).
8. Essayvraag:
Beschrijf het nut en de noodzaak van pointers. Beschrijf het verband met 'references' en 'arrays'. Leg ook uit waarom het gebruik van pointers gevaarlijk kan zijn.

Andere mogelijke vragen

1. Wat zijn de specificaties voor de "case labels" waarop "geswitched" kan worden? Geef voorbeelden.
2. Wat is het verschil tussen de formele declaratie en de actuele oproep van een functie? Geef een voorbeeld.
3. In C++ geldt de "One-Definitie rule": je kan geen tweemaal iets definiëren. Leg duidelijk uit waarom, eventueel geïllustreerd met een voorbeeld.
4. Wat zijn de syntactische verschillen/overeenkomsten tussen een struct en een klasse?
5. Waar moet je bij debugging van je programma in het bijzonder op letten?
6. Hoe kan je een stream output aanpassen aan je wensen? Geef een paar voorbeelden hiervan.
7. "class Shape" is een abstracte klasse. Wat betekent dit, waarom is dat zo voorzien, en hoe zorg je ervoor dat een klasse abstract is? Is er nog een manier om een klasse abstract te maken?
8. Indien je een virtuele functie overschrijft, waar moet je de i.h.b. op letten? Wat is het verschil met overloading?
9. Pointers en arrays noodzaken voorzichtig gebruik. Leg uit. Waarom zijn ze dan nuttig bij het programmeren?
10. Wat is een copy constructor? Waarom is deze vaak noodzakelijk in een klasse declaratie? hoe kan je een copy constructor aanmaken (exacte syntax niet noodzakelijk), en waar zorgt die essentieel voor?