



OPGAVE 4

JOSÉ LAGERBERG - ROBIN DE VRIES - JETSKIE BEKS - MARCO BROHET

Dr. Quackerjack

ARCHITECTURAL SUBPRIME DEBTS



Deadline: zaterdag 30 september 2017, 23:00

1 Leerdoelen

Als je deze opgave af hebt, kun je:

- code debuggen en leesbaar maken,
- bekende algoritmen begrijpen en implementeren,
- goed commentaar geven op deze algoritmen,
- een gebruikersconsole maken met goede invoerafhandeling,
- omgaan met een kleine database.

2 Introductie

In de afgelopen drie weken heb je een introductie gekregen op het gebied van programmeren met behulp van de programmeertaal Java. Zoals je vast al gemerkt hebt, ligt de nadruk niet alleen op het kunnen verzinnen van een oplossing voor allerlei problemen, maar ook op een goede programmeerstijl met commentaar dat het begrip van je code versterkt.

In deze opgave zul je merken waarom deze laatste twee punten ook belangrijke eigenschappen zijn voor een programmeur op academisch niveau. In tegenstelling tot de hieraan voorafgegaande opdrachten hoef je niet zelfstandig de code voor een probleem te bedenken, maar krijg je die aangeleverd. Eén probleem: er is amper sprake van een programmeerstijl, het compileert niet en de functionaliteit is onjuist geïmplementeerd.

Aan de hand van een casus word je beproefd op bovenstaande leerdoelen. En vergeet niet: **onderdeel van deze opdracht is het opzoeken van de gebruikte algoritmen, het correct implementeren ervan en dat je ze begrijpt en uitlegt in je commentaar**. Daarnaast beoordelen we je code alsof je het zelf hebt geschreven. Tijdens het nakijken word je geacht de gemaakte keuzen voor ontwerp, naamgeving en programmeerstijl te kunnen beargumenteren. **‘Omdat het zo in de aangeleverde code stond’ is geen geldig argument**. Enige uitzondering zijn de algoritmen die genoemd worden; deze mag je wel zonder problemen gebruiken.

3 De casus

Dr. Quackerjack, een moleculair biologe, is bezig met een onderzoek naar de zeer gevaarlijke *vogelgriep*. Om dit te bewerkstelligen heeft zij met haar team DNA-monsters genomen bij zowel dode als levende eenden. Het DNA van deze monsters is gesequenced, met als gevolg korte strings als ATCGCTGTA, die ze op hun computer hebben opgeslagen. Hierna heeft dr. Quackerjack geprobeerd een Java-programma te schrijven om het DNA te analyseren, en om relaties te kunnen vinden tussen de opgeslagen sequenties. Het doel hiervan is om betere vogelgriep-detectiemethoden te ontwikkelen.

Omdat dr. Quackerjack geen ervaren Java-programmeur is en ook nog eens te kampen heeft gehad met een corrupt geraakte SSD, werkt het programma helemaal niet en is de code vrijwel onleesbaar geworden. Aan jou de taak om haar programma te herstellen. Zorg ervoor dat er gebruik wordt gemaakt van een mooie stijl, dat er voldoende commentaar aanwezig is (met in ieder geval een beschrijving van de gebruikte algoritmen), dat alle functionaliteit naar behoren werkt en dat de invoer op fouten wordt gecontroleerd.

Dr. Quackerjack heeft helaas geen tijd om persoonlijk te corresponderen over de code. Ze hoopt dan ook dat je zelf uit kunt vogelen wat er gedaan moet worden. En het heeft haast, want op dit moment vliegt een enorme zwerm geïnfecteerde vogels richting Nederland, die naar verwachting deze zondag onze grenzen binnenvliegt. Heb jij de skills om voor de deadline dr. Quackerjacks programma af te maken en zo Nederland te redden?

4 De opdracht

Begin met het leesbaar maken van het programma voordat je het compileert. Als het in zo'n staat is dat je het kunt lezen, begin dan het compileren van het programma. Debug het door van boven naar beneden door de foutmeldingen te lopen en de fouten te verbeteren. Misschien is het handig om bij te houden wat je allemaal hebt gerepareerd, zodat je daar bij latere errors niet meer naar hoeft te kijken. Let wel goed op: dr. Quackerjack kan een paar haakjes en accolades vergeten zijn, en de indentatie is ook niet al te best.

Als alle foutmeldingen weg zijn, kun je kijken naar de algoritmen en de rest van de functionaliteit. Dr. Quackerjack had veel plannen voor dit programma, maar ze heeft ze niet allemaal geïmplementeerd. De voorbeelduitvoer geeft aan wat jouw versie van dr. Quackerjacks programma moet aankunnen.

Zorg er ook voor dat fouten in de invoer goed afgevangen worden, zodat het programma niet kan crashen. Ook is het belangrijk om goed naar de algoritmen te kijken: zoek op internet op hoe ze werken en wat ze precies horen te doen. Vergeet ook niet om goed commentaar bij de functies en algoritmen te zetten. Zo kan dr. Quackerjack nog wat van haar fouten leren.

5 Inleveren

Lever `Opgave4.java` in op Canvas, laat vooral de `.class` file achterwege!

A Voorbeelduitvoer

```
$ cat input.txt
ATCGAC
ATTGCC
CGAATC
$ java Opgave4
Welcome to DNA Matcher v0.1

console> help
LIST OF COMMANDS
    list                print database
    add ...             add to database
    compare ... ..     compare two strings
    read ...           read from file and add to database
    retrieve ...        find in database
    remove ...         remove from database
    quit               stop
    help              print this text

console> add CGAATC
console> add AGAGTA
console> add CACTGA
console> list
CGAATC
AGAGTA
CACTGA

console> read input.txt
Warning: 'CGAATC' already added, skipping.
console> list
CGAATC
AGAGTA
CACTGA
ATCGAC
ATTGCC

console> add ATTGCC
Warning: 'ATTGCC' already added, skipping.
console>compare AATACG ATTGCC
0 1 2 3 4 5 6
1 0 1 2 3 4 5
2 1 1 2 3 4 5
3 2 1 1 2 3 4
4 3 2 2 2 3 4
5 4 3 3 3 2 3
6 5 4 4 3 3 3

Difference = 3

console>retrieve GAGAGA
Best matches:
2      AGAGTA
3      CACTGA
4      ATCGAC
```

```
console> remove CGAATC
console> list
ATCGAC
ATTGCC
AGAGTA
CACTGA

console> add CGAATC
console> list
ATCGAC
ATTGCC
AGAGTA
CACTGA
CGAATC

console> remove ACTACT
ACTACT not in database, nothing done

console> remove
Skipping...
console> compare AGTAGT
Skipping...
console> add
Skipping...
console> quit
```