Blue Minev

228089922

CI505 Functional programming

Reflective report on the development of a software system

In the lab you will examine the assessment brief in detail and start preparing for coursework development.

1) Download and read the assesment brief carefully, noting the deadline

2) Using the requirements, start drafting data structures to be used in the project

3) Create a Word document that you will populate in the labs, which will form the basis of your coursework submission.

4) Consider how to implement each requirement and start making notes into the document.

5) What are the main differences between OOP and FP? How is the solution going to be different when using FP over OOP? Answers to these questions should help you structure your solution.

1. Explain the potential benefits and limitations of the functional paradigm.

2. Solve problems using standard functional techniques and abstractions, such as recursion and the use of higher-order functions.

3. Write programs in a functional style.

4. Use a functional approach to solve modern software problems, such as testing, and/or in domains such as web frameworks, distributed systems and parallelisation.

Define student: id, name, course and module, year

Multiple modules to one course

Marks of the modules

List/ dictionary maybe??

<https://www.geeksforgeeks.org/arraylist-of-arraylist-in-java/>

{[studentID,studentFName,studentLName,yrOfStudy,course,[[moduleID,mark], [moduleID,mark], [moduleID,mark],]], [studentID,studentFName,studentLName,yrOfStudy,course,[[moduleID,mark], [moduleID,mark], [moduleID,mark],]]}

Separate it to 2 lists ??

[studentID,studentFName,studentLName]

[studentID,yrOfStudy,course,[[moduleID,mark], [moduleID,mark], [moduleID,mark],]]

New student: generate id , give name, give course, give year, required modules, choose other module.

Delete student, give id as confirmation, search algorithm and remove?

Query- advanced search, with multiple variables and can be sorted based on a variable.- functions wrapped in functions?

# INTRODUCTION

# DEVELOPMENT

After looking through the requirements for this project I decided to start by tackling the problem using a object oriented approach so that I could understand the scope and requirements better. After learning about records and the stream API in lectures I was then able to confidently start this project using functional programming. By using the knowledge I already had to try and tackle the problem I was able to understand what is needed and issues that I may come across better than if I had started using functional programming. For example before learning about records I was planning on using a list or a dictionary but was encountering a lot of issues due to the fact that each student would need a lot of information and I would essentially have to be using 3D arrays to get around this issue as I knew that objects would not be allowed. When we were taught about records this solved this issue.

A record is the functional programming equivalent to an object its aim is to remove unnecessary boilerplate code such as getters and setters. Records are an easier to implement data carrier class than objects requiring little to no extra methods to set them up.

A computer screen shot of a program code

Description automatically generated*the original data values in my student record*

When considering what data I wanted to have assigned to each student I I knew that there must be a unique value for each student to make it easier for searching students with same names or finding details of the student when you only have this value. I chose to use a unique student ID starting from one and incrementing with each additional student. I also decided to split the students name into a first and last name that way if a potential user only knew the first name of the student they were they would be able to still search for that student. I also decided to include year of study and the course that they were on as data values. I Also included modules and marked for those modules as paired data values for the student.

I decided to use an enum for the course after beginning with a string value to prevent errors when a user would add a student for example misspelling the course name or capitalization issues.

I encountered many issues in working out how to implement the marks and modules section for each student. This is since I did not fully understand the map data value. Originally, I fought the map.of() created a single map and you had to have a list of the maps this meant that when I tried to access the values in this list of maps I've struggled a lot. After some more investigation I discovered that is singular map.of() would suffice as you could set create multiple values inside the map. I still then encountered issues when I try to use this stream API to only get the marks out of the map especially since my original aim was to get an average of these marks. Since I was unable to do this using the stream API I decided to rethink the way I was going to get the value for the grades. Since each student had three modules and therefore 3 marks instead of adding up all of the Marks and dividing it by three I decided to triple the grade percentages and use that instead . For example instead of 70 being a first I tripled that and therefore the requirement to get a first was 210 marks this removed the need to do division altogether and I still got the same results as I would have if I divided the values.

After creating the base of my student record I then began to explore the stream API and its abilities in terms of queries for this student record. I began to think about what queries would be most common when using a student record as well as being able to show my knowledge of stream api. I decided on showing all students ordered by last name, being able to find a student by their student ID, being able to find a student with just the 1st letter of their first name, being able to show all students on a course ordered by grade, showing all students grouped by their grade, and showing all students grouped by their year of study. I also created the option to add and remove students.

When removing a student I decided that the added security you would have to confirm that you wanted to remove that specific student this involved. I decided that the process to remove a student would be for the user to enter the student ID of the student they wanted to delete then the programme would ask if they wanted to remove this student and show the student record for to start student if the user replied yes then the record would be deleted if not the record will not be deleted and I've away a confirmation of what happened would be shown.

# CRITICAL REVIEW

# CONCLUSION

# ESTIMATED GRADE

# REFERENCES