Cover Page

Final Assessment Name: Jonathan Taylor StudentID: 20865991 Practical class: Thursday/12pm/314.218

Introduction

When testing this, please go into the folder code to run the code and the JUnit test folder to successfully see that they do work.

I have choosen to implement the modules listed below:

1. Convert a given String to upper case or lower case
2. Identify whether numeric values are in a given string
3. Identify whether a given String is a valid number or not
4. Remove any numerica values in a given String and then convert the string to upper or lower case
5. Converting a number which represents a length given in meters to feet and vice versa and centimeter to inches and vice versa

The use of a kanban board in conjuction with a work breakdown structure it made completing this assignment alot easier to read and understand. The completion of small tasks eventually all accumulated and in no time, I had completed the detailed decription section.

As I was the only person on this project, I firstly understood the requirements of the assessment, and created a plan that would help me design and implement the software and its modules in the easiest way possible.

Througout this assignment, have written production code, created test cases and implemeneted it using JUnit. The production code has been written to be tested using black box testing and choosen submodules, using white box testing. My submodules either read in a file and wrote out to another, or took in import parameters or would take in user input. The first four submodules, I split into two and two. I made two submodules take in user input and the other two submodules take import parameters, being hardcoded in the main. Using the last submodule, I wrote a scenario below, describing the principles of ethics and professionalism.

Module Descriptions

I choose functionality (a) of category 2, as a result of different countries using different measurment systems, imperial and metric. Thus the creation of a software that can flawlessly do that calculation, can improve the efficiency of projects requiring that.

Preliminary description of your modules:

1. Converting a given string to upper case or lower case a. This submodule, StringConverter() can use default import parameters or user input. For this specific modules user input will be used. User will be prompted to choose between a lowercase or uppercase conversion. An "IF" statement is used to execute desired code of lowercase or uppercase conversion. User will be prompted to enter a word User input will be imported as a parameter then the word is converted to desired format. Then the result will be outputted to the terminal.

2. Identify whether numeric values are in a given string

```
 a. This submodule, IdentifyNumbers() can use default import parameters or user input. For this
specific modules import parameters will be used. The module will import in a string as a parameter
and will identify numbers in the string, by placing each character into an array. Then a for loop
will check each index value, if it is a number or not and isolate them. The number/s will be
returned to main and printed in the terminal.
```

3. Identify whether a given string is a valid number or not

```
 a. This submodule, StringValidation() can use default import parameters or user input. For this
specific modules user input will be used. User will be prompted with a message to enter a string,
with our without numbers. It will run through an If statement and print to the terminal the entered
input if contains numbers. Does not contain numbers then prints, not a valid numerical string
```

4. Remove any numeric values in a given string and then convert the string to upper case or lowercase

```
 a. This submodule, removingNumbers() can use default import parameters or user input. For this
specific modules import parameters will be used. A string will be imported in containing numbers,
the code will remove the numbers. This is done by placing the individual characters and letters
into an array and replacing the numeric values with " ". Once the word/string will be converted to
lowercase or uppercase. Result will be printed to the terminal.
```

5. Converting a number which represents length in meters to feet and vice versa and centimetre to inches and vice versa

```
 a. This submodule, conversions() will use a read file and write to file method. This will read a
file and print results to another file. The conversions are placed in a menu options between m to
ft and cm to inches, where the user is prompted to choose the desired conversion, either feet to
meters or centimetres to inches.  The results of the conversion is then printed to a new file.
```

Modularity

Checklist: **bold text** 1. Does this method have high coupling? 2. Does this method have low cohesion? 3. Is there any redundant code in the method? 4. Are global variables being used inefficiently? 5. Is a control flag being imported? 6. Does the method accept different data type? 7. Are sequential tasks being used in the method? 8. Does the class/methods have a common task with one another? 9. Is there any repetitive code in the submodules? 10. Are submodules free of differing data imports? 11. Are any sequential tasks being used in the submodules? 12. Is the program free of global variables? 13. Do submodules return their values instead of storing them in global variable? 14. Are the submodules only performing one task? 15. Can any of the submodules use another method in its submodule?

I have created a menu that allows the use to test my production code that I have written. The menu consists of options 1-5 and 0 to exit the program. The user will be prompted to choose and option. IF statements are utilised to run desired code based on the choice of the user. For example, if the user choose option 1 then, the test data is batman3, it will then output the just 3 to the terminal. As option 1 of the menu is to identify the numbers.

I wanted to break up the software into self contained pieces such as methods and functions. Above is a checklist used to improve the modularity of my production code. Looking over my production code following this checklist the issues that I refactored most was redundant code, and should be able to be seen in my gitHub log. However my code in utils.java regarding the calculations, may be seen as redundant code however, I had to make sure that each submodule only did one task. It may be seen that a control flag is used in isNumeric() however this does not create high coupling between two different/random things within the method. It is merely checking the input to print a certain statement. Overall it may be seen as a control flag but it has no negative affect on coupling.

image of control flag in code

My submodules, have high cohesion and low coupling as I have split up the software into self contained pieces. Thus high cohesion will lead to greater efficiency in the use of mental resources. I have not used any global variables but have use global constants. This is okay as global constants cannot be changed by a function and will not affect other submodules whereas global variables can be changes by functions and will ultimately affect other submodules.

My submodules are free of using different types of data, as many would import a string value or read in values from a file all of the same data type.

Black-Box test cases

| isNumeric() test cases |

category	test data	expected result
input is a string of numbers	"5991"	"true, input is valid"
input is a string of chars	"batman"	"false, input is invalid"
input is a combination of string + numbers	"taylor22"	"false, input is invalid"

| convertString() test cases |

category	test data	expected result

category	test data	expected result
input is lowercase	"taylor"	"TAYLOR"
input is uppercase	"TAYLOR"	"taylor"
input contains a lowercase char	"BATMAn"	" "
input contains an uppercase char	"Batman"	" "
input contains a number	"5991"	" "

| removingNumbers() test cases |

category	test data	expected result
str contains a number	"TAY22"	"tay"
str contains a number	"tay22"	"TAY"
str contains no numbers	"TAYLOR"	"taylor"
str only contains numbers	"5991"	" "
str contains a number	"spiderman3"	"SPIDERMAN"

| identifyNumbers() test cases |

category	test data	expected result
numID is a string of numbers	"5991"	"5991"
numID has no numbers	"taylor"	" "
numID contains a number	"taylor22"	"22"
numID contains a number	"spiderman3"	"3"

| conversion1() test cases |

category	test data	expected result
file contains data in metres	file.txt, 67.9	222.779
file contains data in ft	file.txt, 99.3	30.266

| conversion2() test cases |

category	test data	expected result
file contains data in inches	file.txt, 69.9	177.546
file contains data in cm	file.txt, 43.5	17.125

I choose to use test cases as they are good for repeatability and test automation. Testing with automation makes the process easier in terms of repeatability. Black box testing in specific, the use of hard coding, makes the process of testing less painful. Most of my submodules used Equivalence partioning, as the input will test a certain behaviour of the software and the expected result will be compared. Boundary value analysis was not used as my submodules, import strings and not numerical values.

White-Box test case

| isNumeric() WB test cases |

Path	Tesst data	expected result
Enter try statement	"5991"	"true"
Enter catch statement	"batman"	"false"

Path	Tesst data	expected result
Enter IF statement	"5991"	"input is valid"
Enter ELSE statement	"batman"	"input is invalid"

| convertString() WB test cases |

Path	test data	expected result
Enter IF statement	"taylor"	"TAYLOR"
Enter ELSE IF statement	"TAYLOR"	"taylor"

I choose the isNumeric() and convertString(), as they take in user input. With whitebox testing I was able to see the production code and make changes if the test was to fail, I would be able to easily rectify the problem. User input would have to go through paths to use the executed code, such as an IF statement or try catch statement. Shown in the test cases above

Test Implementation and execution

I used JUnit and VScode, to complete this assignment. It allowed for easier testing of my test code. JUnit test code differs from normal test code, as there is a change in syntax and different assertions. For exmaple the use of AssertEquals instead of Assert.equals for java test code. The formation of the code, in java it is the "expected", the method call and test data, then message. Whereas JUnit, for example AssertEquals, message, expected, method call and test data.

comparison of test code without JUnit comparison of test code with JUnit

Modeule name	BB test design(EP)	BB test (BVP)	WB test design	EP test code (implementation/run)	BVA test code (implementation/run)	White box testing (Implementation/run)
isNumeric	done	not done - BVP only applies to numerical imports	done	done	not done	done
ConvertString	done	not done - BVP only applies to numerical imports	done	done	not done	done
removingNumbers	done	not done - BVP only applies to numerical imports	not done - not choosen	done	not done	not done - not choosen
identifyNumbers	done	not done - BVP only applies to numerical imports	not done - not choosen	done	not done	not done - not choosen
conversion1	done	not done	not done - not choosen	done	not done	not done - not choosen

Modeule name	BB test design(EP)	BB test (BVP)	WB test design	EP test code (implementation/run)	BVA test code (implementation/run)	White box testing (Implementation/run)
convresion2	done	not done	not done - not choosen	done	not done	not done - not choosen

Version Control

To start this project by creating a local and remote repository and read over the requirements of assignment. After creating the repositories, I require a minimum of 4 branches for, production code, black box and white box testing then test implementation. These 4 branches is where most of the work will be done as the master branch will hold the final code for each section. I would typically want to try and merge at the end, once all code is finished. I prefer to merge my sub branch with the main branch once all the work has been completed. However all working on one branch at a time.

git log excerpt, not the whole log

Ethics

bold text Scenario: You are developing a software that is used for Space craft launching. The company has many research facilities around the world, meaning facilities within the United States compared to Australia would use the imperial system. The research is done in Australia thus calculations are done in meters. The launch is done be performed in the US as Australia lacks the resources to do so. The software created is to convert meters to feet. Once completed the US scientists use the software and find faults in the conversion, being one decimal off. The software has rounded the numbers too early before the final calculation, which can cause critical rounding errors in the final result. As a consequence the men aboard the spacecraft may be place in a fatal position.

bold text a. Discuss how lack of ethical and professionalism can result in harmful effects using the code you have designed and implemented. You may refer to lecture 9, seven points useful in identifying ethical issues, as a guidance.

Acting with ethics is a result of professionalism, as it is assumed that doing a job "properly" results in working with ethics. However ethics are subjective and depends on the context of what is right and wrong. A lack of ethics is present, the faulty software has affected the safety of the "public", specifically the men who are aboard the spacecraft. Due to this the software developer's integrity may come into question, resulting in a termination of duties. The developer has put the "public" lives at risks. As with calculations done within space, a minor rounding error can be costly, thus the criticality of rounding increases. The developer of the software has not acted ethically as this mistake has caused harm to the "public" and may affect the reputation of the company in addition to great financial losses, resulting in becoming unable to perform another launch. The principle of deontology, is a duty or obligation that is upheld by software engineers. The developer of this conversion software had an obligation to the company and the men boarding the spacecraft, but due to this mistake it would portray the developer as negligent. Thus the integrity of the developer is compromised. In reference to consequentialism, the developer is acting unethical, as they did not understand the consequences of their actions to the greatest degree of accuracy due to the context. The programmer for the software is a new developer that has little experience with big companies. Thus once the developer is tasked with the job, they are offered a bonus of $15 000 when completed within a specified time limit. The developer completes it in time. Believing their code is flawless. The developer was too focused on the bonus and did not execute proper error checking. This is a clear conflict of interest as he is prioritising his personal gain over the safety of the "public".

bold text b. Using ACS or IEEE-CS Ethical guidelines, give two suggestions to avoid ethical and professional issues in your software proposed in this assignment.

To avoid the conflict of interest that can be seen above; personal gain over the safety of the public. In the ACS code of ethics, the first value is primacy of public interest. The developer should be placing the interests above those of personal gain/interests. To combat this the developer may undergo various types of self-reflection and morality reflection. The integrity of the developer and the code have come under question, a lack of professionalism and incompetence, as they have ruined the reputation of the company. Employees could look at and memorise moral exemplars that are made specific to their company, to act in ethically in the eyes of their employer. For example the ACS guidelines hope to improve professionalism, enhancing the integrity of the company. As each person may have their own morals and may justify all their actions as ethical, it becomes essential that things such as the ACS guidelines exist. These are a set of guidelines for the employees to follow. If an employee followed the guidelines it can result professional development, meaning greater professionalism thus working in a way that is ethical.

Discussion

The overall assignment was a time consuming process, from the testing of code and report writing, it took some time to complete this. I believe that my work is simple but efficient it can execute what is needed by the user. The production that I wrote is one of many ways I believe it can be written, but I took a kanban board approach to this assignment. I created my plan of what I needed to do and so on. Much like the lectures, I would only allow three cards per section, however nearing completion I started to have a pile up of tasks in review, as I had not properly error checked them. Once an error chheck was completed, I created a new Kanban board with the bugfixes listed in to do. This made the whole assignmnet more efficient and allowed me to complete the assignmnent in a timely manner.

The downside of my production code is that the first 4 functionalities take in string inputs, thus I could only use Equivalence partitioning testing and not boundary value analysis. As BVA only can be used when numerical value imports are taken in. That is the shortfall of my assignment.

I think i could better improve on the production code to be more specific. if I had created a software requirements specification (SRS), it may have helped in narrowing down what the actually production code should be doing and how the user will interact with it. Within this I would of created user stories, identifying what the user could do and what they would gain from it.

This is a learning experience of how actual projects would happen within a work environment, thus errors and mistakes will be made. But it can be used to improve my project management in the future by utilising the tools that have been equipped with.