Student Number: 17823420

Individual Report CI701

Introduction

The purpose of this review is to explore the use of programming languages in software development and describe the sections of the ATM application that I worked on. From my own personal experience, I will be talking about Java, Python and web development languages such as HTML and CSS. The aim of this review is to analyse the strengths and weaknesses of these languages and the learning resources available online to a beginner in the programming world. I will be looking at the readability of the syntax and how easy it is for a new user to understand it, if it can be used across multiple platforms and if it has a large community support with resources to help troubleshoot problems.

Strengths

Python

Python is a great beginner language due to its syntax and readability. It has an extensive number of libraries such as NumPy which is used for processing numerical data and scientific computations and scikit-learn which is used for machine learning tasks [1]. There are also libraries that can be used in web development including Django and Flask [1]. I have personally used the OpenCV library in the past for computer vision tasks compromising of tracking human faces and hands for personal projects and I found it easy to learn due to the extensive amount of resources online.

Java

Java is a powerful programming language that has a relatively simple syntax which makes it an accessible choice for most new programmers [2]. There are a wealth of online learning platforms and tutorials that teach Java for beginners, the language will run on nearly all systems from Mac and Linux to android. Java is an object-oriented programming language which allows its code to be organised and maintained easily for larger code bases and projects. It also has better security built in with features such as encapsulation and inheritance [2]. Java is also very similar to other well used languages such as C++ and C#.

HTML and CSS

HTML is a widely supported language by every browser and is extremely easy to learn. It is also lightweight making it fast to load and can be integrated easily with CSS of JavaScript [3]. You do not need a dedicated IDE to edit it meaning that you can code anywhere with it. Another advantage of HTML is that it is completely free, and developers do not need to pay for licenses [3].

The same goes for CSS, it can be edited anywhere but unlike HTML, it is a more difficult to get a good grasp of it. There are many things that can be down with CSS from animations to styling and even responsive design to allow your content to adapt to different screen sizes [4]. This is crucial given that most people visit websites from their phones over computers or laptops. A lot of the animations you see on websites can be done with CSS only, not necessarily with using JavaScript.

Weaknesses

Python

Python is known for being a slow language when compared to others that are pre-compiled such as Java and C++ [1]. This is because python is an interpreted language which means each line of code is executed one at a time by the interpreter [1]. Python is also less secure than other languages as given its extensive libraries, not all of them may be properly screened and can harm a user's computer [1]. Another drawback of python is its bad memory management. Due to its interpreted nature, it uses more memory than other languages and this could be an issue in more resource restricted devices such as mobile phones [1].

Java

Java lacks in the GUI department, it is not the ideal language to create native looking applications and has an outdated look to its GUI library [2]. If Java is being used in a commercial setting, then there are licensing fees that businesses need to pay which might not be desirable for smaller businesses with a limited budget [2]. Like python, Java can suffer from performance issues due to its virtual machine and memory management [2]. This is not a huge problem but requires expertise to mitigate these issues, something a beginner would not know how to do straight away.

HTML and CSS

HTML is a very simple language to grasp but by itself, does not create dynamically beautiful websites. For that, the user must learn CSS and progress to learning the more difficult JavaScript [3]. The structure of a complex website written in HTML is very hard on the eyes. There are a lot of div sections, and it can get quite confusing if you are coming into a new project that has already been written [3].

CSS looks simple but, it is quite a difficult language to master. All the different animations and dynamic effects one can do is not so easy once you break it down. There are a lot of things to learn, and this can be quite overwhelming for beginners like me. During my web development coursework, I found that modifying layouts in CSS was not straight forward at all. Elements do not do what you expect them to do, and styles can cause conflictions with other styles as it is a 'cascading' language. There is also the added element that you end up writing too many useless lines of code that affect the performance of the website [4].

Recommendations

Based on the research done in this report, each language has its role and place in the software development world. Python is the go-to language for data science, be it computer vision, machine learning or computational analysis. However, it is not good for making games or web applications given it is a very slow language.

Java is a tried and trusted language. There are lots of resources for it on the internet and companies would rather use something that is safe instead of taking a risk using new technology. It is also platform independent, meaning it will work on most, if not all devices. Java can be seen as an all-rounder language, it can be used to make GUI applications like Adobe Acrobat Reader, it can be used to make games like Minecraft and even to make computational software like

MATLAB. It also has a lot of overlap with other popular languages like C++ and C# making it a very good language for a beginner to learn.

HTML and CSS are the standard for web development. There are other packages and libraries that can be added onto JavaScript that can further enhance the look of your website such as React and Three.js, although, the foundations of your website will nearly always be built using HTML and CSS.

Group Project and my Contributions

Scene Builder and CSS

I decided to use scene builder instead of writing the layout by hand because of its simplicity. In my spare time outside of university and for some light preparation for the course, I was following along to Java tutorials on YouTube and the topic of using JavaFX with scene builder was presented to me. I found this tool very intuitive and easy to use hence why I suggested we use it for this project.

The idea for the layout of the GUI was inspired by a real ATM you have buttons on the sides of the screen or sometimes they are touchscreen. You also have the numbered buttons on the bottom with only ‘clear’ and ‘enter’ on either side. It would have been better to have just one input and output screen where the user enters their details and amount, they wish to withdraw and then it updates on a single screen. This would have been a lot neater and more presentable I feel.

The CSS file was very short and simple to write. Having practised writing CSS for the web development course, I found this very straightforward as I only needed to change the background colour for the inputs and the application itself, the text colour and some of the button colours such as the ‘enter’, ‘clear’ and ‘exit’ buttons. In addition, I used a software I purchased for my hobbies called Aseprite to draw the icon for the application. This was entirely optional, but I felt the application needed an icon and not just the default JavaFX icon on the taskbar.

This section of the project was probably the most relaxing and straightforward section of them all. The worst was yet to come with many more problems and challenges to overcome.

Controller Class File

I contributed to the first half of the controller file which included adding the button objects from the FXML file and creating the action event for the buttons. I then added some more program states and specified the default state of the application. To use objects from the Bank and Bank Account class files I declared them before starting the work on the methods that would control the logic of the program.

I started by making an initialisation method that would load some methods that would give the application its default look. This included the display message method, loading the trial accounts and setting the default state.

The handle enter method is the largest of them all. It utilises the programs 'states' as the events for each case in the switch statement. Which state the application is in will dictate what happens when the user presses the 'enter' button. I first needed to take what the user entered and assign that to the variable named 'input'. The first state is to prompt the user to enter their account number or press the 'add' button to create an account. If they choose to enter the account number and press enter then it will take them to the next state which is to enter their password. Once they have entered their password, the 'findAccount ()' method is called which looks to see if there is an existing account that matches the number and password. If there is a match, then it will go to the logged in state. If a match is not found, then the user will receive an error message, and the state will be reverted to the default state so they can try again. From here the user can check their balance, withdraw money and deposit money. Once they are happy, they can either log out or exit the application entirely.

If the user presses the 'add' button, then they will be prompted to enter their name and an if statement will check if that name is only made up of letters. If numbers are found, then an error will pop up and the user must try again. If there are no numbers present, then a new bank account object is created with empty fields. A random number generator will generate an account number and password between 1000 and 9999 and present that information to the user. Once the user is happy, they only need to type '1' and press enter which will take them to the account display state. Once again, an if statement checks if the input entered by the user is a number and equal to '1'. If so, then they will be taken to the logged in state and can use their account.

Bank Class File

The bank class file contains the ArrayList that stores the account details of the users. I chose an ArrayList as you do not need to specify the number of accounts prior to running the application. The' addAccount ()' method creates a bank account object with four parameters and adds the account to the ArrayList. The find account method uses and enhanced for loop that will look for an account with matching number and password when the user logs in. It then returns a Boolean value of true if it finds an account or false if it does not.

The 'setAccount ()' method also uses an enhanced for loop and is used to set the account in the logged in state if the 'findAccount ()' method finds one. It needs to return an item of the BankAccount datatype so if it finds the one, it will return the account and set it to the current account for the user. If it does not find an account, it will return null.

The final method is linked to the 'handle deposit' method and needs to be created to allow the for loop in that method to access the accounts due to encapsulation. This will allow the application to update and overwrite the previous information of that account when the user changes their balance.

Conclusion

Through this project I have gained invaluable hands-on experience in Java working as part of a team. My contributions involved designing the application layout and user-interface, developing some of the control logic and implementing some of the core functionality of

the system. This project has helped me improve my skills at object-oriented programming, debugging and problem-solving.

There are lots of improvements that can be made to this application such as adding a .php server to store the accounts data, adding a way for the user to change their password in the eventual case that they will forget it and adding a hash map to store the accounts as opposed to an ArrayList for faster indexing. I hope that it can prepare me for the real world of software development and how teams are structured and organised to get something done on a deadline.

References

[1] Y. Gavrilova, “Pros and Cons of Python,” *Pros and Cons of Python*, Oct. 30, 2023. https://serokell.io/blog/python-pros-and-cons (accessed Jan. 2025).

[2] K. Rafalski, “Pros and Cons of Java Development in 2023,” *www.netguru.com*, Sep. 12, 2023. https://www.netguru.com/blog/java-pros-and-cons (accessed Jan. 2025).

[3] S. Das, “Advantages and Disadvantages of HTML - ellow.io,” *Ellow*, Sep. 06, 2023. https://ellow.io/advantages-and-disadvantages-of-html/ (accessed Jan. 2025).

[4] S. Das, “Advantages and Disadvantages of CSS - ellow.io,” *Ellow*, Oct. 20, 2023. https://ellow.io/advantages-and-disadvantages-of-css/ (accessed Jan. 2025).