# ECS414U/A Miniproject form

Queen Mary University of London

2021/22

|  |  |
| --- | --- |
| Name | Hazal Kara |
| Student ID | 210350678 |
| Submitted file name | MiniProject.zip |
| Level of this program (1, 2, 3, Extra) | Extra |

|  |
| --- |
| Brief description of the program. Write the chosen theme and a high-level overview of the features (two or three sentences should suffice). |
| The theme I chose was the investment app and in my GUI I have a quick menu where I give the user the option to add an asset to the arraylist of assets and to deposit based on the amount they have and if the account they want to find is not there I used exceptions to handle that. I also used polymorphism where I overrides the asset currency in the subclasses and used a getter to print that when showing the asset Balance and my superclass is an abstract class which has an abstract method in the subclasses telling the user about the asset type (Crypto or Share). And used File I/O to show the user their wallet balance. I used Swing to add JList and images which is something we did not learn so it was found through research listed in (Other Comments). |

|  |  |
| --- | --- |
| List all your source code files, and briefly describe their roles. Add as many rows as necessary. Mark the main file used for compilation in bold. | |
| File name | Description |
| Account.java | It holds an arraylist of Assets and makes the changes to Assets and changes the balance for the user. |
| Asset.java | It is a superclass which the subclasses Crypto and Share inherit the methods inside and it has many getters so that the class Account can make changes to the arraylist. |
| Crypto.java | It is a subclass that extends to Asset and uses polymorphism by overriding one of the methods in Asset so it can have a different currency from the subclass Share and has a different value to the double value and different type to the string type. |
| Share.java | It is a subclass that extends to Asset and uses polymorphism by overriding one of the methods in Asset so it can have a different currency from the subclass Crypto and has a different value to the double value and different type to the string type. |
| Prompt.java | Prompt makes all the changes visible and makes the submit button and other features work. |
| WindowCloser.java | It closes the window of the GUI in the class InvestmentApp.java |
| **InvestmentApp.java** | InvestmentApp class is the main class where the GUI is created, in this case I used swing so JPanel just incase I wanted to add elements for extra marks and this class makes the changes in Assets arraylist by having an Account initialised and using it to deposit into my assets and adding them and withdrawing from them. It also has a file read and write method which writes to AccountBalance.txt and reads from it and is reseted each time the GUI opens. I also added something extra which uses swing and its JList and ImageIcon which is some JLabels that add a list to add the assets instead of typing it in and when clicking on the asset you want on the side panel is shows the asset name again, the type and the stocks graph. |
| AccountBalance.txt | This has the Account wallet balance and we use the methods in InvestmentApp to change the value and reset it and read it. |

|  |
| --- |
| Class diagram, in the format specified in the instructions. |
| class InvestmentApp extends JFrame{}  class Account{}  abstract class Asset{}  class Share extends Asset{}  class Crypto extends Asset{}  class Prompt extends Frame{}  class WindowCloser extends WindowAdaptor{} |

|  |
| --- |
| Usage instructions. Describe briefly what features are available to the user and how to use them. If File I/O is used, list and describe the files involved. |
| There are 7 features available to the user:   * One of them prints the assets list which basically just lists the assets the Account has. * The 2nd shows the Account wallet which needs to be above or equal to the amount you want to deposit into the asset and this feature uses file I/O to read the wallet amount from AccountBalance.txt. * The 3rd feature is adding Assets which opens a JList and when you click on the asset you want it adds it and shows information on the side panel (stock graph, name and type of asset) and you cannot add the same asset twice until it is removed. * The deposit button Is the 4th which asks you to input the name as it is, the type of the asset and the amount you would like to deposit and also uses file I/O to update the wallet in AccountBalance.txt since to deposit you withdraw the amount from the wallet to the asset. * The 5th is depositing into Account which uses file write to change the wallet amount the user has in AccountBalance.txt. * The 6th feature is the withdraw from asset button which also prompts you for the asset name, type and amount and then withdraws and if the balance falls to 0 the assets is deleted from the buttons and also uses file I/O as it deposits the amount withdrawn which is usually less in AccountBalance.txt because when investing there is a chance of losing an amount and this shows that. * The last feature it the withdraw from the wallet feature which also uses file I/O as it changes the wallet amount (AccountBalance.txt). This is all in the class InvestementApp.java. |

|  |
| --- |
| Other comments. |
| I used Swing to add some extra features like my JLists and ImageIcon (for my stocks graph) which I had researched all on my own and found most of how I could do the stuff implemented in it using stack overflow and YouTube tutorials and oracle helped me understand how and why it works and helped me further implement it so the samples I found fit my code better. You should open the JList to full size to see the graphs when first selecting. I also added colour to my project so the background colour is pink.  These are some sources I used to implement it into my code:   * <https://stackoverflow.com/questions/6714045/how-to-resize-jlabel-imageicon> * <https://docs.oracle.com/javase/6/docs/api/javax/swing/ImageIcon.html> * <https://stackoverflow.com/questions/3775373/java-how-to-add-image-to-jlabel> * <https://docs.oracle.com/javase/7/docs/api/javax/swing/JList.html#setValueIsAdjusting(boolean)> * <https://docs.oracle.com/javase/6/docs/api/javax/swing/event/ListSelectionEvent.html#getValueIsAdjusting()> * <https://stackoverflow.com/questions/646288/why-do-jlist-selections-occur-twice> * <https://github.com/BranislavLazic/SwingTutorials/blob/master/src/main/java/JListTutorial.java> * <https://stackoverflow.com/questions/7257942/java-gui-programming-setting-the-fore-background> * https://www3.ntu.edu.sg/home/ehchua/programming/java/J4b\_CustomGraphics.html#:~:text=3.1%20java.,-awt.&text=The%20class%20java.-,awt.,LIGHT\_GRAY%20%2C%20ORANGE%20%2C%20and%20PINK%20.   **Compilation Command:**  As this project uses a package in order to run follow the following steps:  In your terminal change directory(cd) to the folder called MiniProject.  Then you javac Project\InvestmentApp like this:  **Execution Command:** java Project.InvestmentApp.  It should look something like this. |