Object Oriented Programming

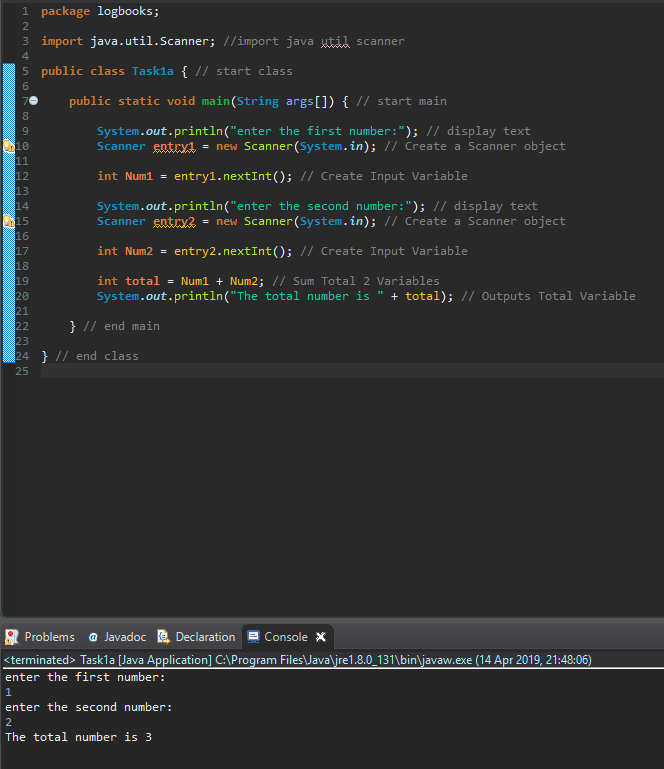
MOD004881

Submission date: 09/08/2019

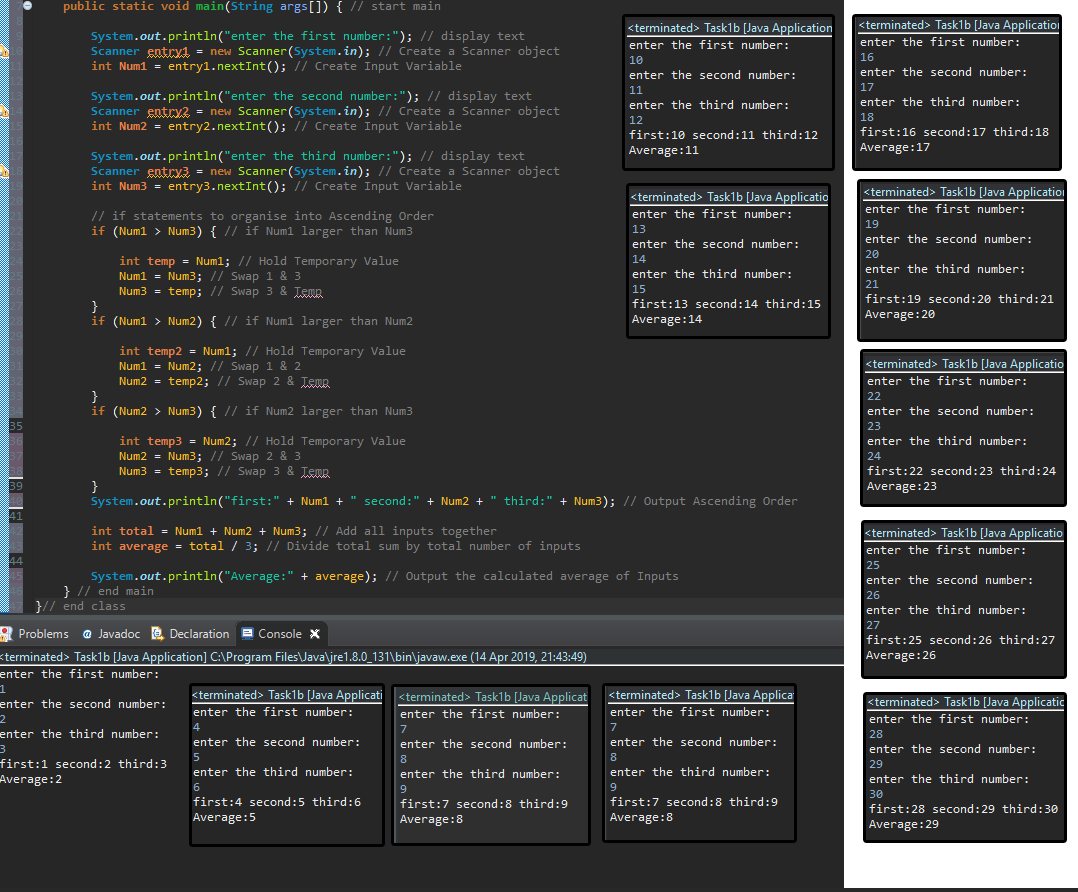
Assignment 1 (element:010)

From: 1625601  
To: Dr Ioannis Kypraios

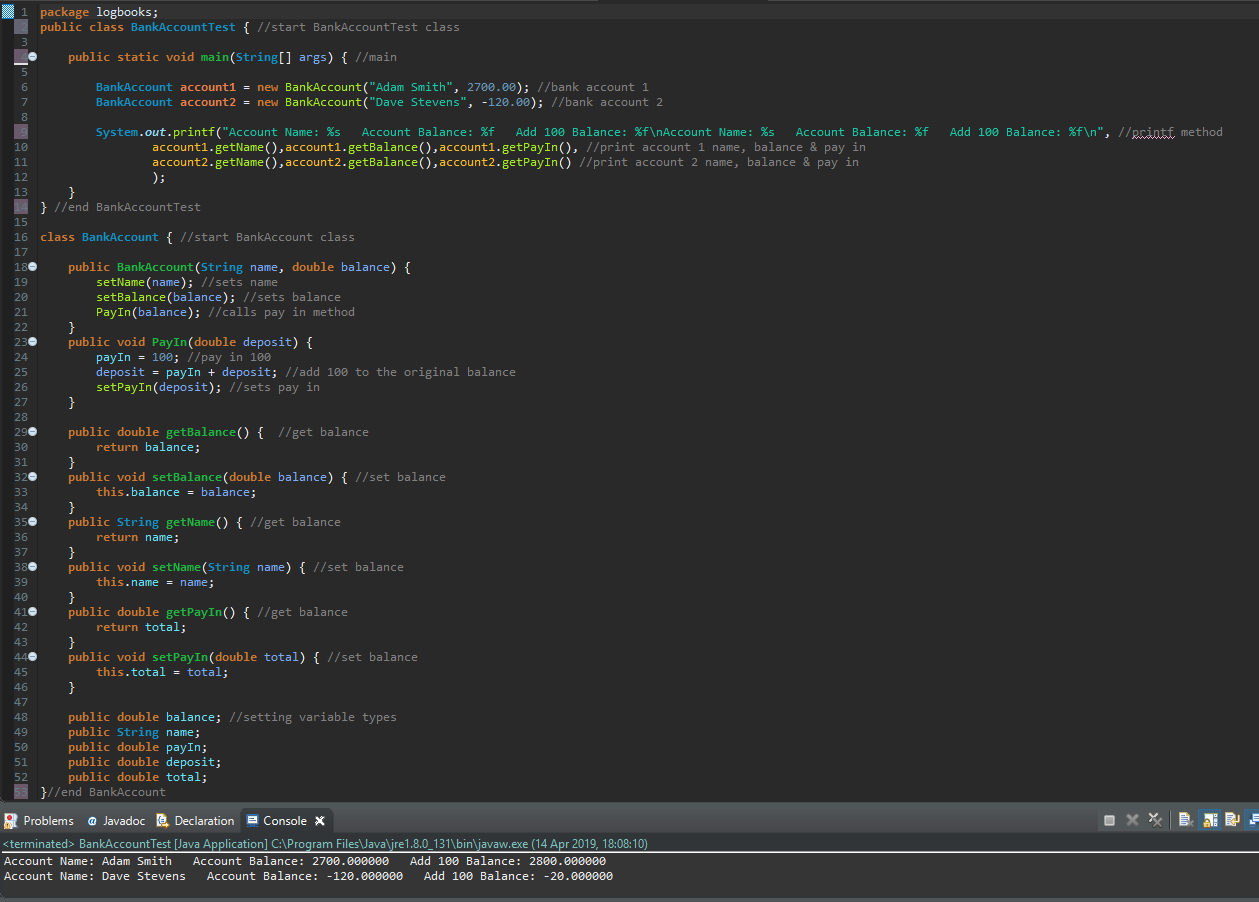
Task 1



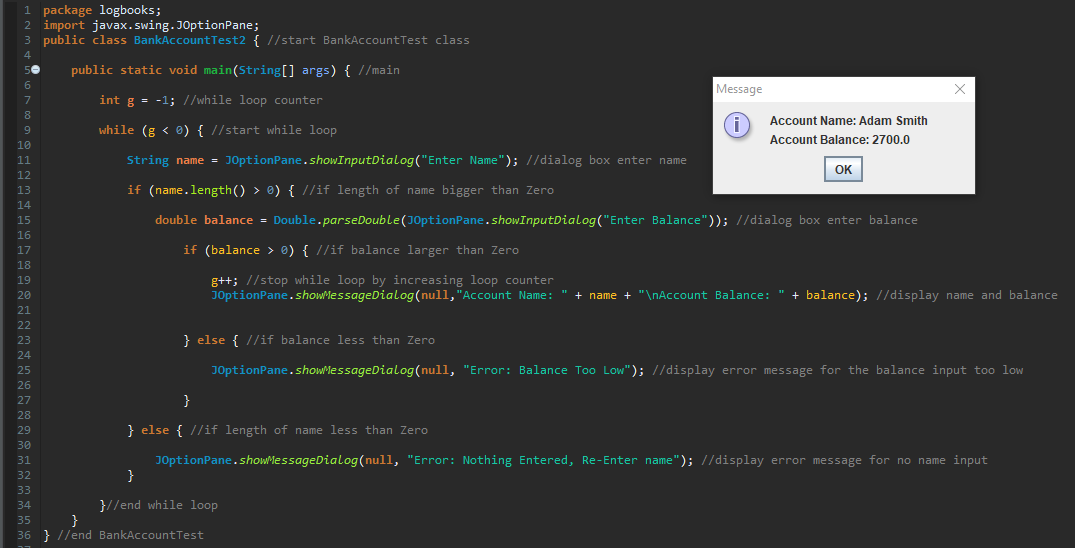
This code uses the imported java util scanner which allows you to input data into the console. This code asks the user to input 2 numbers and adds them together as a total at the end. The system asks for the first number to be entered then the second and assigns those inputs to integers. Once assigned both the integers are added together to make another integer called total and outputs the result into the console.

Task 1

This code now asks the user to input 3 numbers, the code arranges them into ascending order and also calculates and average of the 3 numbers entered. To arrange the numbers into ascending order it uses the method of holding temporary numbers, if number 1 is larger than number 3 swap them, if number 1 is larger than number 2 then swap them and finally if number 2 is larger than number 3 swap them which then arranges the numbers into ascending order. To work out the average the code adds up the 3 numbers that have been entered then divides them by 3 to give the mean average of the numbers entered.

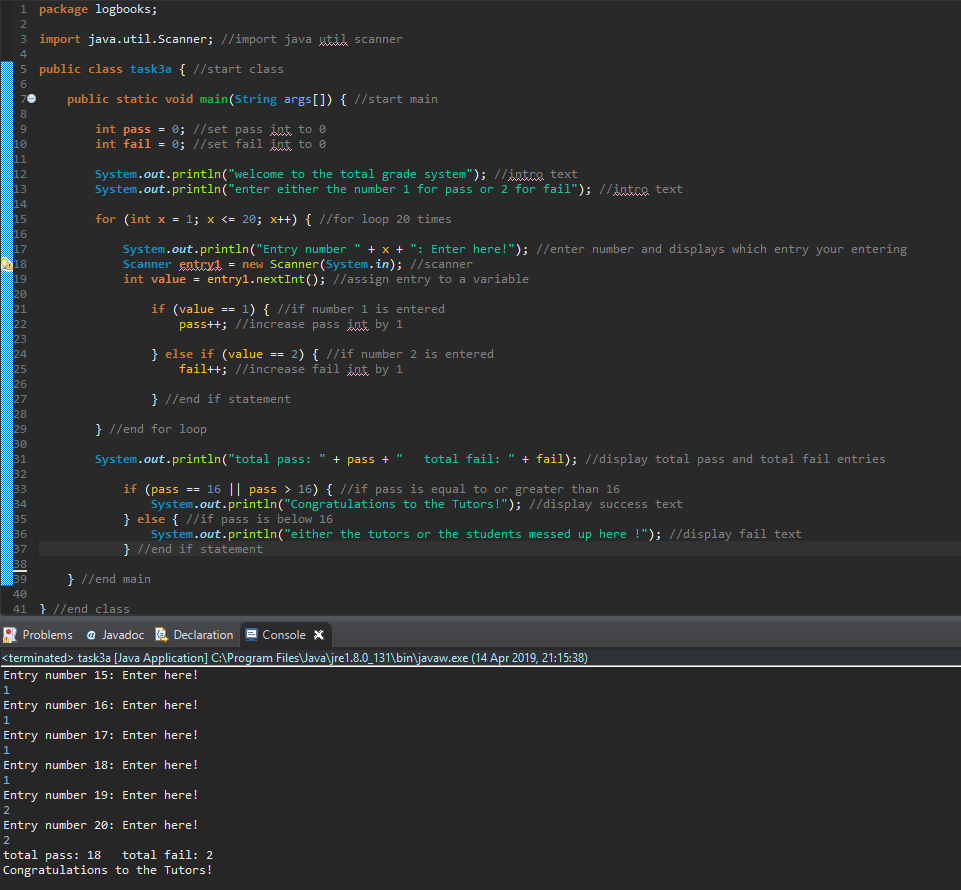
Task 2

This code is a virtual banking system with a name and balance of a user implemented at the top. Then from the BankAccountTest main class calls the BankAccount class to use get and set methods to output results into the console. It uses the printf declaration to print the info into the console which also makes the code more efficient. It uses a getName() and getBalance() method which is called in BankAccount to be able to output the information to the console by using setName() and setBalance(). getPayIn() has been put in place with a variable to add 100 to the original balance of the account which runs an algorithm in the PayIn method and then is called from the BankAccount method to output the results into the console.

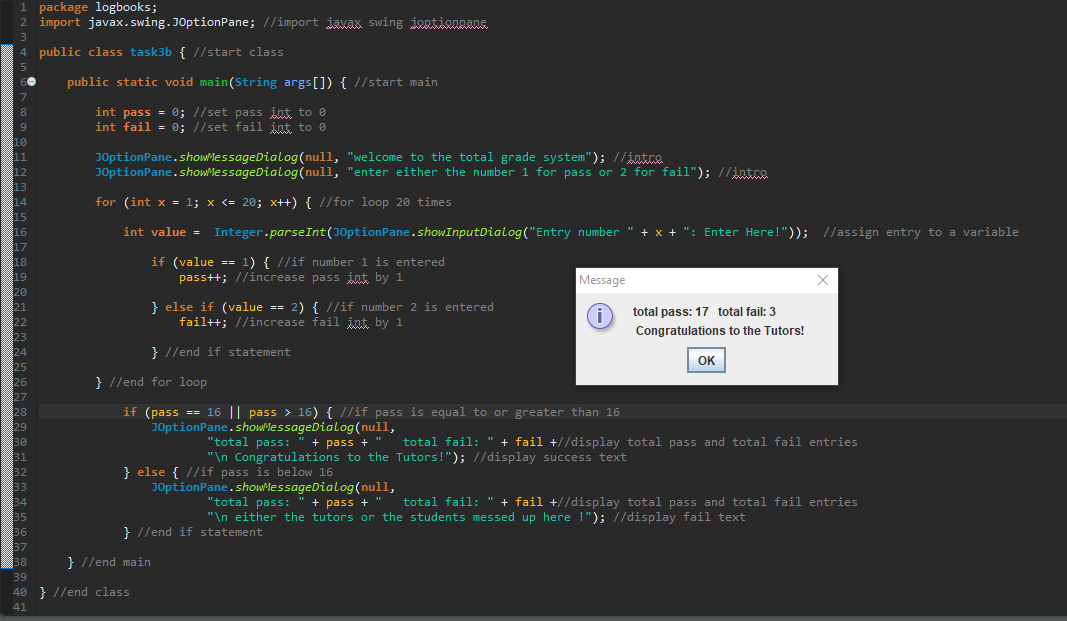
Task 2

This is a newer development of the BankAccount code which allows a user to use input dialogs to enter a user’s name and bank account balance. Instead of using the console System out method this code has imported the javax swing joptionpane to be able to use dialog and message boxes. The first input dialog asks for a name to be entered and assigns it to a string then the second box asks for a balance to be entered and assigns it to a double. This runs on a loop and it has a successful input so if nothing has been entered into the name box is displays a message dialog to enter a name, if the balance entered is below zero it also displays a message dialog stating that the balance is too low so the user will have to enter a valid name and a valid balance to be able to enter the details into the system. Once correct details have been entered a message dialog displays the account name that has been entered and the account balance.

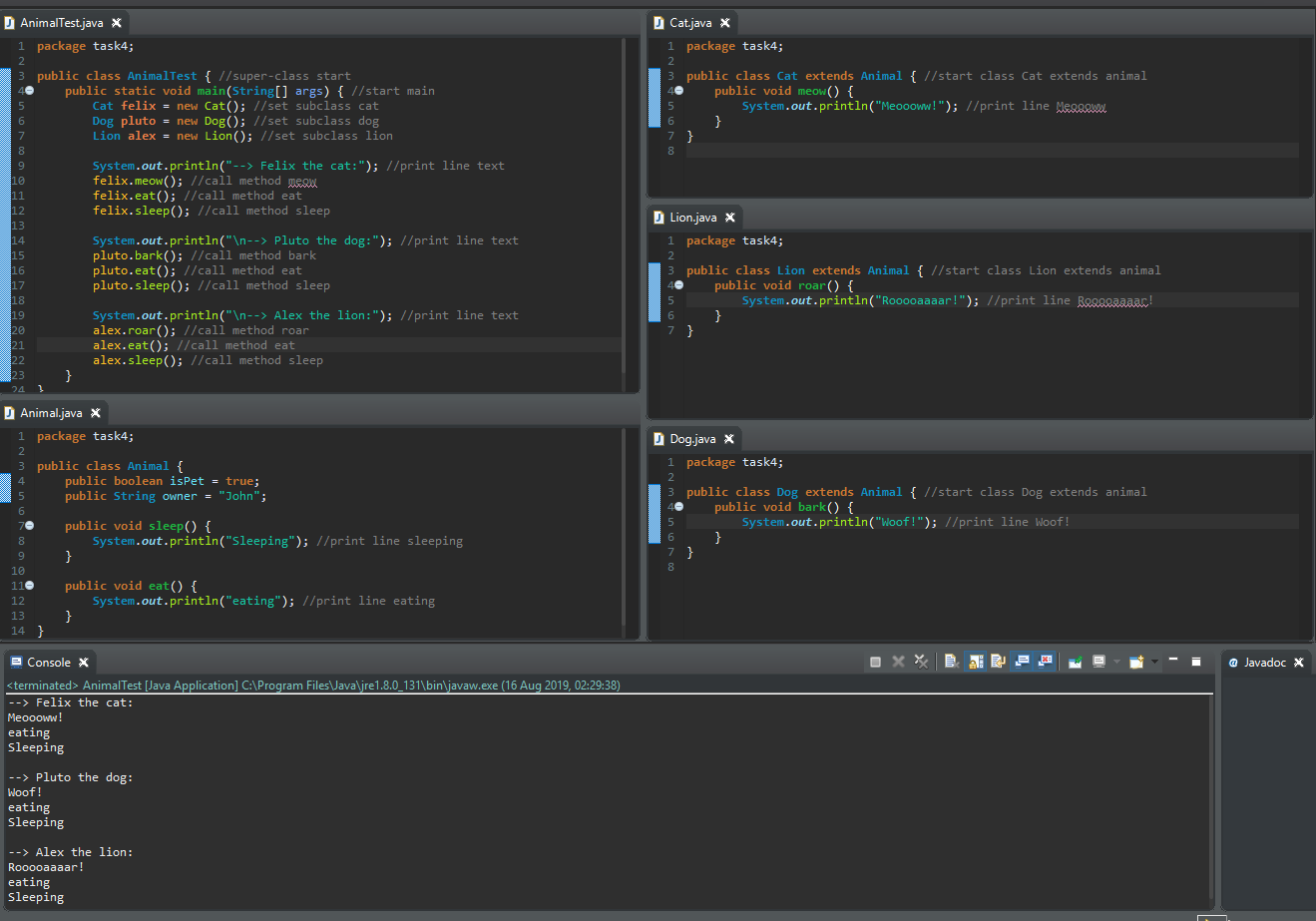
Task 3



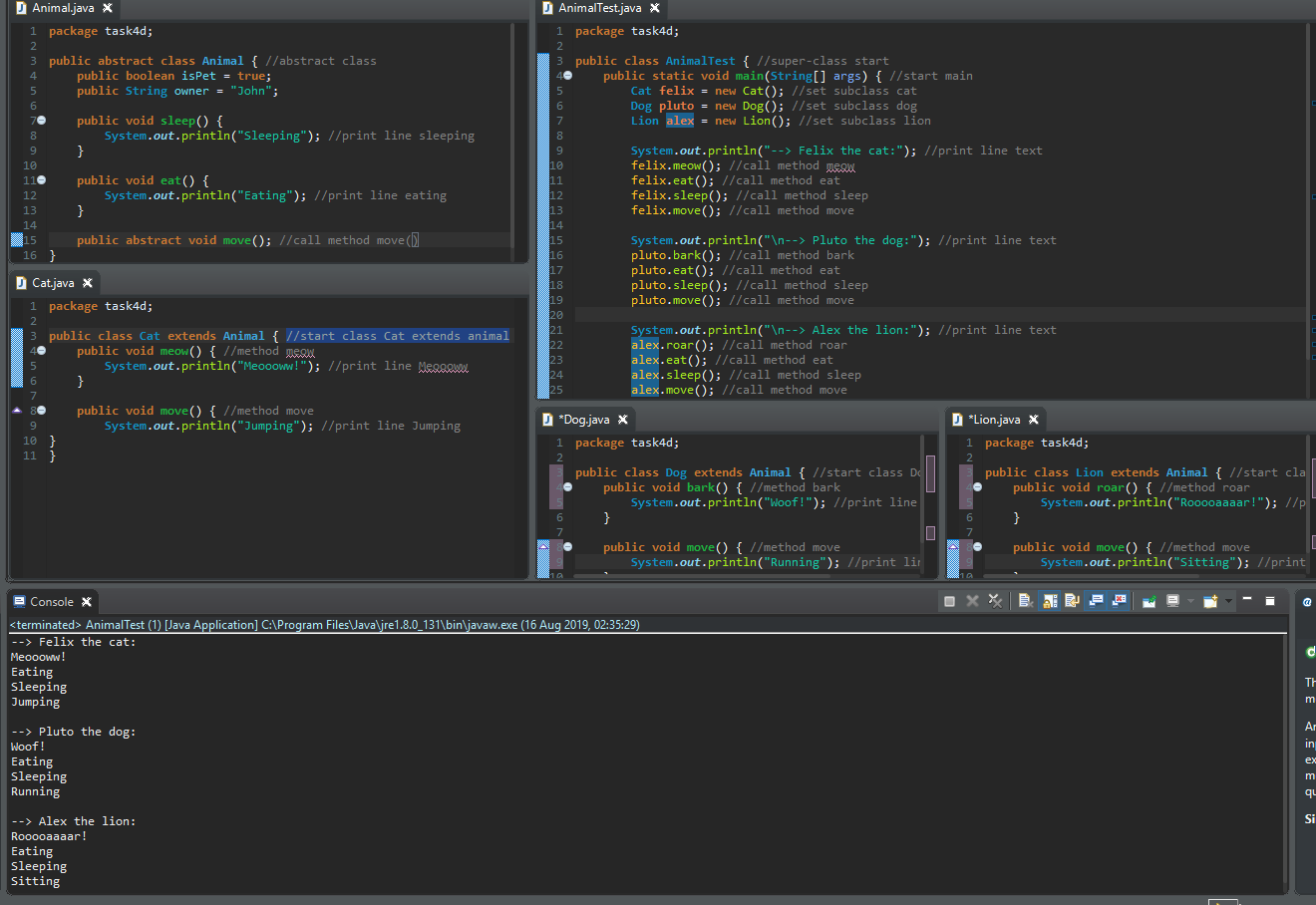
This code allows a user to enter either a 1 for a pass or a 2 for a fail to represent a grade pass or fail. It uses the System out method which uses a for loop to be able to gain 20 entries from the user, every time the user enters a 1 or a 2 it will either increase the pass integer in increments of 1 or increase the fail integer in increments of 1. One the loop has completed the code outputs the total passes and total fails into the console as well as a message saying “congratulations to the tutors!” if the pass rate is greater than or equal to 16 out of 20. if the pass rate is less than 15 the console message says “either the tutors or the students messed up here !”.

Task 3

This code is the same as before and loops 20 times asking for then number 1 or 2 to be entered to represent a pass or a fail. But instead of using the console it uses the javax swing joptionpane to use message dialogs and input dialogs instead. Just like the console version of this code the pass and fail integers have to be declared as 0 before being able to use them as counter. Every entry from the user in the dialog box it states which entry number you are entering between 1 and 20 . Once all 20 results have been entered the if statement has had to include the total pass and total fail results in each if statement so that the statistic information and the message at the end could be on the same message dialog. The message at the end still remains “Congratulations to the Tutors!” if the pass rate is greater than or equal to 16 and “either the tutors or the students messed up here !” if the pass rate is less than 16.

Task 4

This code uses a superclass and calls apon sub-class to gather data by using methods to gain the information to bring forward into the main class to be able to display the information into the console. In this example it uses different types of animals, uses the names of those animals along with calling the method to pull data from the correct class.

Task 4

This code has been adapted from the task before and takes advantage of the abstract feature to go back and forth between subclasses to gather information to be displayed onto the console. Taking the Cat Felix as an example it calls for an extra method called move() which has to go through the class cat which extends into the Animal class, which is then declared move() as abstract so it would need to include the sub-classes for the method to be output into the console.