

COMP-8702: Virtual-Zoo Assignment Report

Task-1 Q&A:

Q1. What issues (if any) did you encounter with the program?

After working on the code, I didn't face any issues with the existing code but with my code some general issues like typo errors missing words. The main-method initializes a Virtual-Zoo class and triggers the start of the simulation by invoking the beginSimulation() method. The VirtualZoo class serves as a program for simulating a virtual zoo. Nevertheless, the code solely comprises the displayWelcome() function, which handles the task of displaying a welcoming message on the console.

Q2. Describe any issues with the code as initially downloaded.

I have not faced any compilation error with the given code initially, As that code is only limited to certain part and also we need to write the BeginSimulation() as it is not partially written there are no errors. As there is also no scanner class it doesn't read any user input. I didn't face any errors after completing the following program also.

Q3. what did you have to fix in order to compile and run the code successfully in IntelliJ?

As mentioned before there is no errors with initial code, However I have faced errors with my own methods where I have to change the way of approach for getting end output. while applying conditions on if else method I faced few difficulties however I have fixed them by applying multiple conditions nearest to the output and applied the out-put which have worked at the end.

Q4. Explain how the first line of the provided toString() method works.

The toString() function can be quite valuable in presenting the state of an animal object in a well-structured manner, whether it is being printed to the console or showcased in a user interface. It is used to show user in a simplified manner too. In the initially presented code, the toString() method overrides the default toString() method of its corresponding class. It returns the string representation that encapsulates the object's status, encompassing details like its names, species and few other mentioned objects

IsDead() method is used to create a method that displays animal name when the animal is dead, As per the rules of zoo code there are certain limitations on what to give and what not to give again if we break those rules animal will die and displays "Animal Name" (Dead).

Task-2

Q1. How did you ensure that the user provided a positive integer for the number of animals?

I have created a method **Num_of_Animal()** to read user input and check If the user entered a negative number, it asks user to enter a positive integer and if user inputs an invalid integer it prints out to user to re-enter a valid number. It also reads the condition whether the integer is greater than 0 or not to ensure user enter positive integer.

Q2. How did you ensure that the user provided a unique name for each animal?

I have created a method **Animal_Name()** which make sure that the user input's different names for the different animals by registering the names that has already been entered in the *usedNames* list. To check the duplicates of names I have created a while-loop and an inner loop the prompts user to enter a name and checks if the name exists in the *usedNames* list. If the user entered name already exists in the list the then the system gives the output that name already exist and the animal's name prompt appears again. If the name doesn't match the saved list, then it adds the names to the list.

The loop makes sure the that it prompts the question add name until he enters the unique name.

Q3. How did you ensure that the user provided a valid species for each animal?

I have created a method **Animal-Species** that makes sure the user enters a name that is from the list of species we mention by using switch-case.

I have created a loop, where the user is asked to enter species for the animal linked to name of animal. A switch-case statement is used to validate the user input with the list of species used in the pre-existed list. If the user enters a name out of the list, then the user gets a prompt to re-enter name along with the message that prints in-valid species.

Task-3

Q1. How did you check whether an item specified by the user for an animal was different from the most recent item that was given to that animal.

I have created a method **isLast_item** that contains Boolean operators to check whether the current item is different from the last item. The variable is assigned with the condition to compare with the last given item variable in order to check it.

If the variable returns True then the most recent item is same as current item then it prints a message about same item given.

If the variable returns False then the current item is different from the last item where the animal object gets updated and moved to the next step of the process.

Q2. The zoo has asked you to modify the program so that an item can only be given to an animal if it was not the same as either of the previous two items it was given. How would you need to change the code to achieve this?

There are two ways to achieve this one is through using multiple variables with conditions just like previous task lastitem and another one is storing the recently given items in arraylist where we need and check whether the item given is different from the current arraylist and the last_item list is replaced whenever the new item which is different from the list is added.

Task-4

Q3. Describe what difference having the totalCost variable as static makes to your program.

By declaring the **Total_Cost** variable as static within the VirtualZoo class, it becomes a variable associated with the class itself rather than with individual instances of the class. In essence, this means that the **Total_Cost** variable is shared among all instances of the VirtualZoo class. This static declaration facilitates a centralized and consistent tracking of the total cost throughout the zoo simulation, ensuring that cost updates are universally reflected across all instances.

Q4. Describe what changes need to be made to your program in order for the user to manage multiple zoos with their own separate animals and costs.

HashMap can be used to create more than one zoo using the existing objects which has been created for initial project one can achieve this by adding additional names of the zoo to the map and by running simulations.

Task-5.

1. Is there an alternative way that we could determine the species of an Animal object without calling the getSpecies() method?

We can use **instanceof** operator within if-else conditional statement alternatively in-order to determine species of an animal object without directly using getSpecies() method

For Instance:

```
If(object_name instanceof class)
{
//operational statement of class
.....
}
Else (object_name instanceof class2)
{
//operational statements
..... }
Else(condition).....so on.
```

Q2. What impact does making the `Animal` class an abstract class have on our program?

By making the Animal class as abstract, it mandates that subclasses must offer solid implementations for each animal subclass, such as Tiger, Elephant, Rhino, Panda, and Monkey. This requirement guarantees that each distinct animal type possesses its unique implementation of methods like giveFood(), giveWater(), and giveToy(), which are abstractly declared within the Animal class. This approach enables us to establish shared behavior and characteristics within the Animal class, while delegating the specific implementation details to its respective subclasses.

Q3. What difference would be making the `Animal` class an interface, instead of an abstract class, have on our program?

If the `Animal` class were transformed into an interface rather than an abstract class, the resulting program would necessitate animal types to furnish their own implementations for the methods specified in the interface. This alteration would permit multiple inheritance, granting unrelated classes the flexibility to implement the interface. However, it would eliminate any shared state or common behavior among animal types. It is worth noting that this approach would require each animal type to implement all the interface methods, which might lead to code duplication when dealing with shared behaviors.

Q4. The zoo has asked you to write a Graphical User Interface for the program you have written. Could it be used as is, or would you need to rewrite parts of the code to accomplish this? Discuss and explain.

In order to develop a Graphical User Interface (GUI) for the program, certain adjustments would need to be made to the existing code. The current implementation revolves around a command-line interface, which relies on text-based input and output via the console. To facilitate a GUI, the code would need to undergo modifications to interact with graphical components and effectively handle events.

These include selecting a suitable GUI framework, replacing console-based input/output with GUI components, handling user events, and implementing robust error handling and input validation mechanisms to ensure smooth user interactions with the interface.