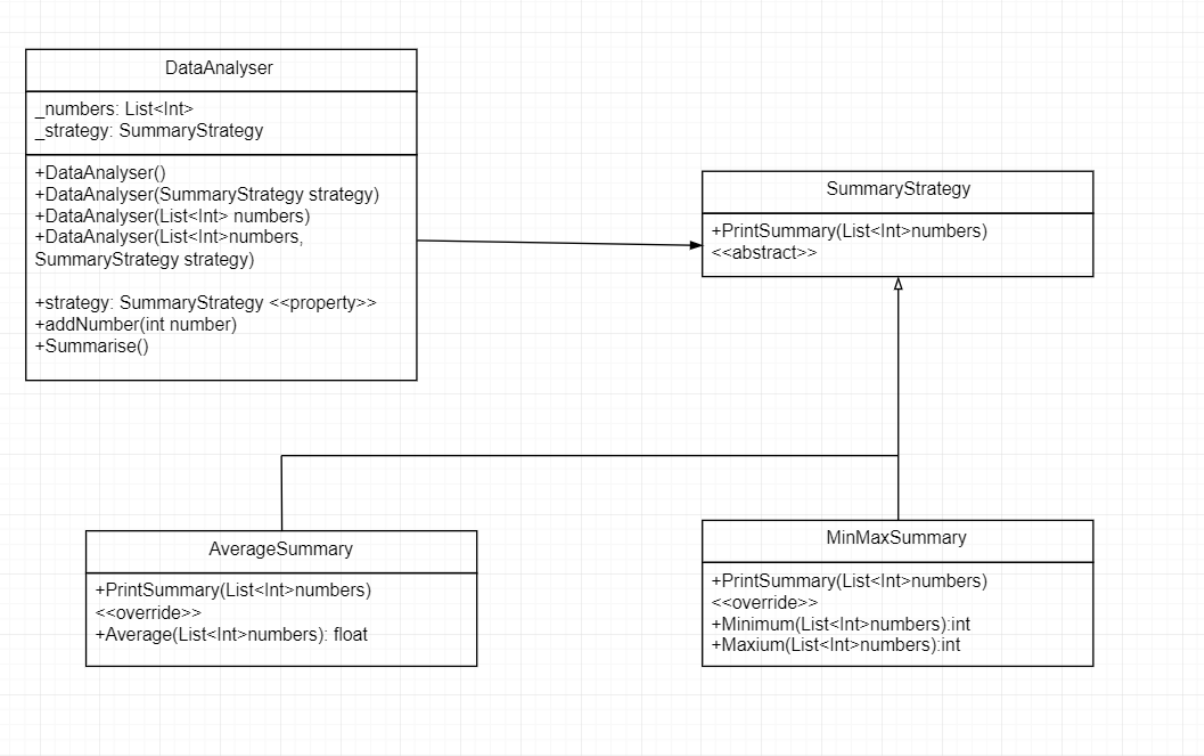
Name: Tran Duc Anh Dang Student ID: 103995439

UML:



Question 1: Describe the principle of polymorphism and how it was used in Task 1.

- The principle of polymorphism describes the concepts of OOP where different types of objects of different types can be access through the interface or abstract class can represent for many different types of objects.

- In Task 1, polymorphism was used in class "AverageSummary" and "MinMaxSummary". They are inherit from an abstract class "SummaryStrategy". Therefore, "AverageSummary" and "MinMaxSummary" can be accessed from "SummaryStrategy". Furthermore, "SummaryStrategy" class may have many different forms, which is an example of polymorphism.

- Coming to the case that when printing out the values in "DataAnalyzer" will becoming much simpler as it only requires to call out _strategy.PrintSummary(_numbers) as "SummaryStrategy" are represent for either AverageSummary and MinMaxSummary in this task.

Question 2: Using an example, explain the principle of abstraction.

- The principle of abstraction is the way to handle the complexity by hiding unnecessary details from the user. Instead of thinking on how it works, the abstraction still enables the user to use the system by implementing inputs. Furthermore, the principle of abstraction is also about representing an entity by removing details but focusing on the aspects of interest, as abstraction allows the user to define the entity or class in OOP without focusing too much on details.

Example: Pilots flying an airplane, they only knows the pushing or pulling the stick will makes the plane going upward or downward but they don't know about the inner mechanism of the plane or the implementation of pushing and pulling the sticks, etc in the plane.

Question 3: What was the issue with the original design in Task 1? Consider what would happen if we had 50 different summary approaches to choose from instead of just 2.

- With the original design, when the program had 50 different summary, its design will require many variables for all summary approaches. Also summarise method will has many same codes for each summary strategy, therefore the efficient of the program will be reduce.

References

Janssen, T. (2017). OOP Concepts for Beginners: What is Polymorphism. [online] Stackify. Available at: https://stackify.com/oop-concept-polymorphism/.

Janssen, T. (2017). OOP Concept for Beginners: What is Abstraction? [online] Stackify. Available at: https://stackify.com/oop-concept-abstraction/.