Royal Melbourne Institute Of Technology

ADVANCED PROGRAMMING FUNDAMENTALS

Assignment 1

Name of the Student 1: Siva Kowshik Sripathi Panditharadyula, s3672957  
Name of the Student 2: Monika Vurigity, s3675394

Contribution of Student 1: 50%

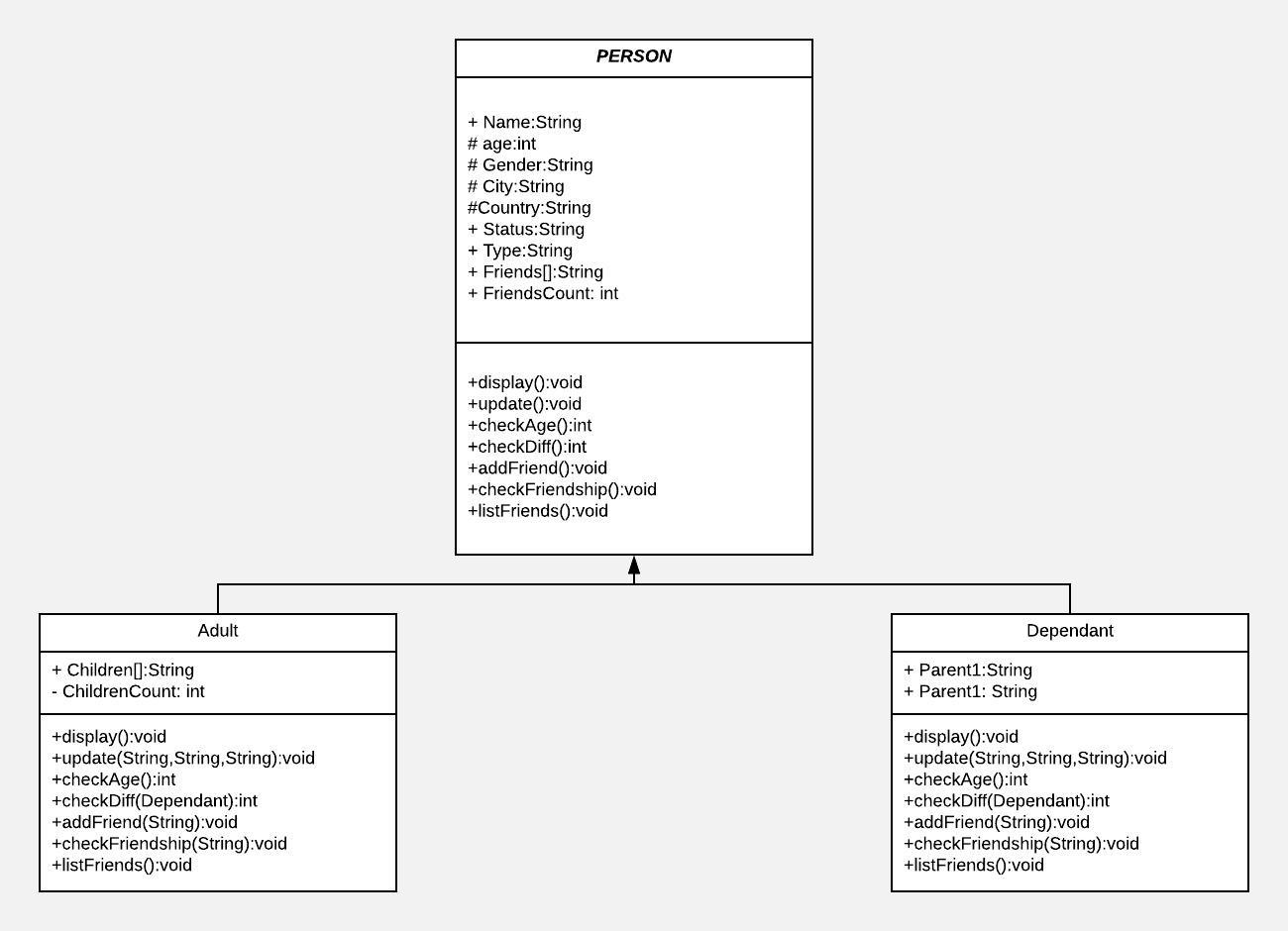
Contribution of Student 2: 50%

# Storing Profile Information

The program uses four distinct classes in order to store the data for each person’s profile. The mininet class holds the necessary methods and variables for the initiation and the functioning of the application. Further, the Person class is an abstract class. It acts as the parent class for the two child classes, namely Adults and Dependent. The Person class keeps all the basic requirements of each profile like name, gender, age, friend-list and so on. The Adult class and the Dependent class inherits from the Parent class all the basic characteristics. The Adult class adds another two important variables that are necessary to keep the records of their children and the number of them. The names of the children, if any, of each adult are stored in the respective array. The Dependent class provides two variables to store the names of the parents for each dependent.

In the mininet class, arrays are made to instantiate each adult and dependent profile. Every time that the user enters new records, new objects are created and stored in their respective class type arrays.

# Class Hierarchy



The class hierarchy as shown in the UML Class diagram helps to build up the network structure for the application. As per the social network policies, the hierarchical structure assisted in segregating the adults from the dependents, in terms of allowing operations and features. Though both inherit the traits from the same parent class, the added data members and member methods help to fine tune the provided features.

# Efficient Connections

The main menu in the mininet class helps to add and select any particular account. On searching for a particular account to be selected, the program enters into a sub-interface, that simulates a personal Profile page of an individual. This assists in updating, deleting, viewing or adding friends to the respective accounts, thus maintaining and expanding the network in isolation.

The program brilliantly uses the hierarchical structure of the class to set up connections between the objects or accounts on the social network. The profiles are created on their respective classes but the Adult Person objects are only restricted to connect or be friends with Adults. This is achieved by looking for names to be friends with, in the Adults array only. Similarly for the dependents. This reduces the search time of the program. The program needs not to look for names that are out of context for the respective person. This enhances the efficiency. Moreover, in order to create a Dependent, the Adult array is searched for the names of the parents that are entered for the dependent. If no such names are found, error messages are displayed. On successful searches, the name of the particular dependent is stored in the array of the concerned adult couple. In addition, the parent-1 and parent-2 variables of the dependent object are updated with the respective names of the parents.

In addition, when deleting an adult, it is checked if the adult has a children or no. If a child connection exists, the deletion is prohibited, as it would require to delete the dependent account as well.

**Bibliography**

Liang, Y.D. and Tsai, M.J., 2013. *Introduction to Java programming: brief version*. Pearson.

Tempero, E., Yang, H.Y. and Noble, J., 2013, July. What programmers do with inheritance in Java. In *European Conference on Object-Oriented Programming* (pp. 577-601). Springer, Berlin, Heidelberg.