Advanced Programming

Assignment 2

s3647369_Yidian He

Explanation of Design

1. **Changes in Assignment 2**
   - New classes are added, comprising the Relationship class to create interface of relations (e.g. friends, couple, parent, colleague, classmate), the AllExcepteion class to handle exceptions, the MainStage class to create GUI and a YoungChild class extends Child class for ease of error handling when define the relation betwenen two people.
   - Variables of parents in the Child and YoungChild class constructor were changed to String. Also, parameters of the methods changed to String object. The reason is that the product needs import data from an embedded HSQLDB database or text files and the messages all that are read from user interface are Strings. Further more, there are no duplicate names in the database so that the String of names can be treated as an identifier. Therefore, using String object as transferring parameters would be more reasonable.
   - To store the person's profile, the map collection was adopted in this assignment as for the String of names can be treated as the key in the mapping and the object Person that contains the profile information is set as value of the mapping.

2. **How the new classes are organized**
   - An Abstract Person class
     - An Adult class extends Person class
     - An Child class extends Person class
     - An YoungChild class extends Child class
   - A Relationship class throws relevant exceptions which aggregated by
     - A Connections class including a connections constructor
     - A Couple interface used by Adult class
     - A Colleague interface used by Adult class
     - A Family interface used by Person class
     - A Friend interface used by Person class while not used by YoungChild class
     - A Classmate interface used by Person class while not used by YoungChild class
   - A Driver class to handle all the actions on the backstage of the user interface.
   - An AllException class to process possible errors handled by the Exception mechanism of Java, which aggregated by
     - TooYoungException, NotToBeFriendsException, NoParentException, NoAvailableException, NotToBeCoupledException, NotToBeColleaguesException and NotToBeClassmatesException classes.
   - A MainStage class to create Graphic User Interface on the forestage of the MiniNet product.
   - A MiniNet class as a start-up class.

3. The process that the program interact with user and external data source

The program would start with importing the data from text files and it would connect to an embedded HSQLDB database if cannot find "people.txt". If failed to connect to either data resource, error messages would display on the user interface which handled by Java exception machanism.

The program would firstly import the relation to the system and then import the data of people to ensure a child or a young child to have parents. All the data would be stored in the mapping for further usage. If the program is connecting to the database, any modification of the person or any new person the user added would update in the embedded database.

On the stage there are 6 buttons for user to choose: Guide; List All People; Add New People; Define Relationships; Find out Relationships and Exit. When user click on the above buttons, the program would interact as below and the actions of the program would display in the console.

- Guide:

  An introduction of how to use the MiniNet system.
- List All People:
  - Display all the people's names in the MiniNet.
  - When clicked on a person's name, the program would show the person's profile. In the profile interface, the user can do the following interactions with the selected person:
    - Delete the person from the net. If the adult have children, the person cannot be removed.
    - Update the person's profile.
    - Find out the person's social connections (including friends, colleagues and classmates lists)
    - Go back to the list of all people.
- Add New People:
  - If the new person's age is out of range, the program would throw NumberFormatException.
  - If the age is from 0 to 16, there would display two extra text fields for user to enter parents name to ensure a child has parents in the MiniNet.
- Define Relationships:

  The user could enter two people's names that are exist in the net and select their relation to define a new relation. The program would search from the 'relation.txt' file to check if the connection can be define. If the two people cannot connect as the selected relation, the corresponding exceptions would be thrown and the error messages would display on the interface.
- Find out Relationships:

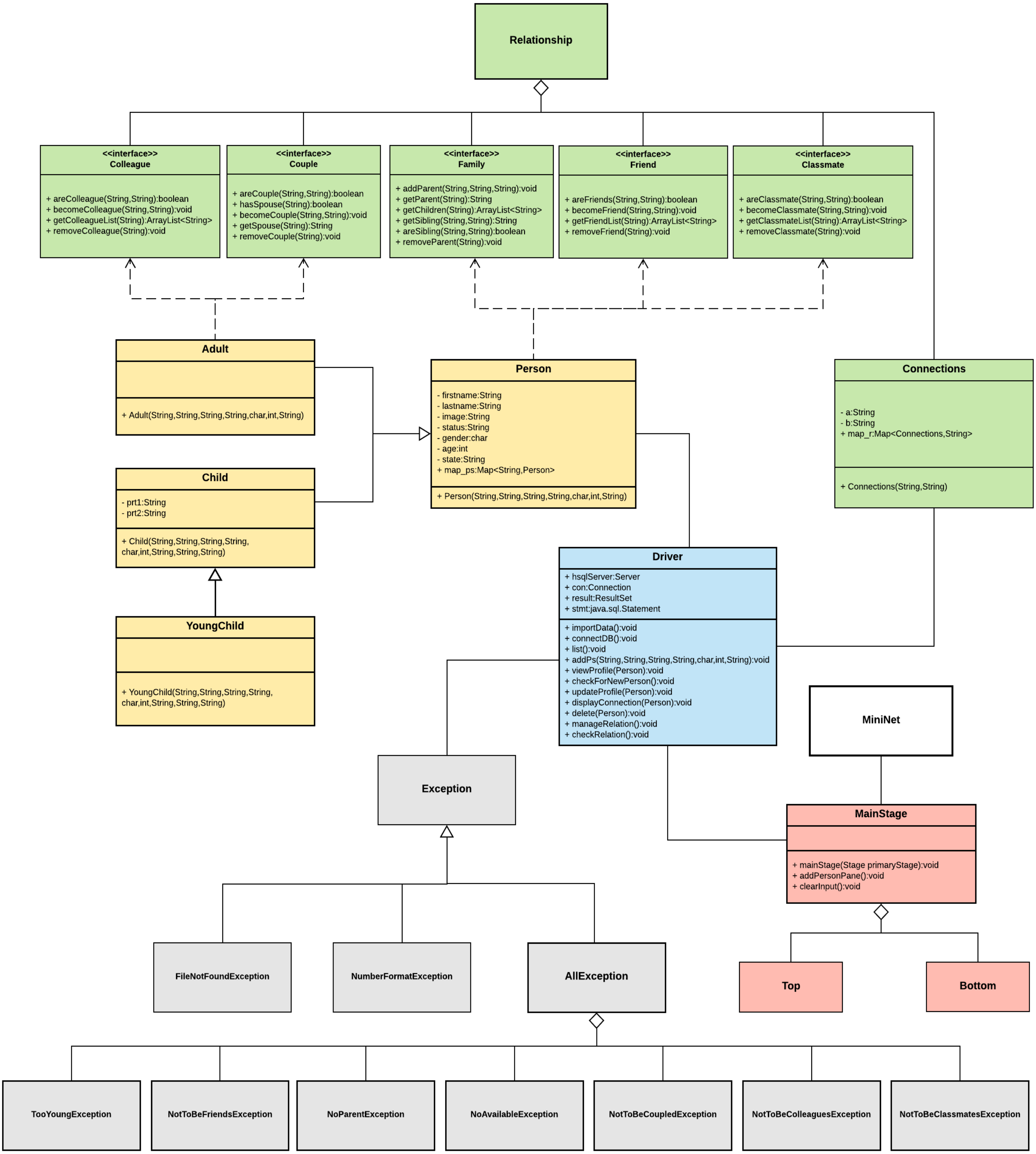
  The user could enter two people's names that are exist in the net and click on the 'check' button, and then the program would call the corresponding methods that return a boolean value and display their connections. If they are not directly connected, it would display 'They are not connected'.
- Exit:

  The program would stop running when the user click on the 'Exit' button or close the MiniNet window. If the program is connecting to the embedded database, this action would call the HSQL Server to stop as well.

Link to GitHub

https://github.com/rmit-s3647369-He-Yidian/MiniNet/tree/branch_ass2

Class Diagram



Relationship
<<interface>>
Colleague
+ areColleague(String,String):boolean
+ becomeColleague(String,String):void
+ getColleagueList(String):ArrayList<String>
+ removeColleague(String):void
<<interface>>
Couple
+ areCouple(String,String):boolean
+ hasSpouse(String):boolean
+ becomeCouple(String,String):void
+ getSpouse(String):String
+ removeCouple(String):void
<<interface>>
Family
+ addParent(String,String,String):void
+ getParent(String):String
+ getChildren(String):ArrayList<String>
+ getSibling(String,String):String
+ areSibling(String,String):boolean
+ removeParent(String):void
<<interface>>
Friend
+ areFriends(String,String):boolean
+ becomeFriend(String,String):void
+ getFriendList(String):ArrayList<String>
+ removeFriend(String):void
<<interface>>
Classmate
+ areClassmate(String,String):boolean
+ becomeClassmate(String,String):void
+ getClassmateList(String):ArrayList<String>
+ removeClassmate(String):void
Adult
+ Adult(String,String,String,String,char,int,String)
Person
- firstname:String
- lastname:String
- image:String
- status:String
- gender:char
- age:int
- state:String
+ map_ps:Map<String,Person>
+ Person(String,String,String,String,char,int,String)
Connections
- a:String
- b:String
+ map_r:Map<Connections,String>
+ Connections(String,String)
Child
- prt1:String
- prt2:String
+ Child(String,String,String,String,
char,int,String,String,String)
YoungChild
+ YoungChild(String,String,String,String,
char,int,String,String,String)
Driver
+ hsqlServer:Server
+ con:Connection
+ result:ResultSet
+ stmt:java.sql.Statement
+ importData():void
+ connectDB():void
+ list():void
+ addPs(String,String,String,String,char,int,String):void
+ viewProfile(Person):void
+ checkForNewPerson():void
+ updateProfile(Person):void
+ displayConnection(Person):void
+ delete(Person):void
+ manageRelation():void
+ checkRelation():void
MiniNet
Exception
MainStage
+ mainStage(Stage primaryStage):void
+ addPersonPane():void
+ clearInput():void
FileNotFoundException
NumberFormatException
AllException
Top
Bottom
TooYoungException
NotToBeFriendsException
NoParentException
NoAvailableException
NotToBeCoupledException
NotToBeColleaguesException
NotToBeClassmatesException