## CS 112 Programing II

## Lab Session 8: Abstract Classes and Interfaces

**The Student Abstract Class**

Student is an abstract class *(the code of this class is given in the following page)*

* **It contains three String properties**
  + nationality
  + name
  + phoneNumber
* **Each of those three properties has an accessor and a mutator**
* **It also includes two abstract methods**
  + acceptName()
  + acceptPhoneNumber()

SaStudent, UsStudent and FrStudent are concrete subclasses of Student

* Each of these subclasses includes a *no-args constructor* which creates an object with the appropriate nationality property
* Each of the three subclasses overrides acceptName(). The implementation of acceptName() in each subclass differs based on nationality:
  + Saudi names normally consist of four parts
  + American, three parts
  + French, only two parts
* Each of the three subclasses overrides acceptPhoneNumber(). The implementation of acceptPhoneNumber() in each of those subclasses depends on the country

### Test class

* declares three objects based on SaStudent, UsStudent and FrStudent classes and stores them in an array of type Student
* for each of those objects, it prints the nationality, and calls acceptName() and acceptPhoneNumber(), and prints the name and phoneNumber

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subclass** | **Nationality** | **Parts of Names** | | | | **Phone Number** | | |
| **Country code** | **Area Code** | **Number** |
| SaStudent | Saudi | First | Father | Grandfather | Family | 966 | 2 digits | 7 digits |
| UsStudent | American | Middle | Last | - | 1 | 3 digits |
| FrStudent | French | Family | - | - | 33 | 1 digit |

1. Create the UML class diagram showing Student, SaStudent, UsStudent and FrStudent and their relationships.
2. Create the Test, Student, SaStudent, UsStudent and FrStudent classes.

**import** java.util.Scanner;

**public class** Test {

public static void main(String[] args) {

Student[] students = {new SaStudent(), new UsStudent(), new FrStudent()};

for (Student student : students) {

System.out.println(student.getNationality() + " student");

student.setName(student.acceptName());

student.setPhoneNumber(student.acceptPhoneNumber());

System.out.println("Name: " + student.getName() + " Phone: " +

student.getPhoneNumber());

}

}

}

**abstract class** Student {

private String nationality, name, phoneNumber;

public String getNationality() {

return nationality;

}

public final void setNationality(String s) {

this.nationality = s;

}

public String getName() {

return name;

}

public void setName(String s) {

this.name = s;

}

public String getPhoneNumber() {

return phoneNumber;

}

public void setPhoneNumber(String s) {

this.phoneNumber = s;

}

public abstract String acceptName();

public abstract String acceptPhoneNumber();

}

**class** SaStudent extends Student {

/\* Write the code for this class here. Please note that the code for this class is   
 very similar to the code for the following UsStudent and FrStudent classes \*/

}

**class** UsStudent extends Student {

UsStudent() {

this.setNationality("American");

}

@Override

public String acceptName() {

Scanner input = new Scanner(System.in);

System.out.print("Enter first name: ");

String firstName = input.nextLine().replaceAll("[^a-zA-Z]","");

System.out.print("Enter middle name: ");

String middleName = input.nextLine().replaceAll("[^a-zA-Z]","");

System.out.print("Enter last name: ");

String lastName = input.nextLine().replaceAll("[^a-zA-Z]","");

return firstName + " " + middleName + " " + lastName;

}

@Override

public String acceptPhoneNumber() {

Scanner input = new Scanner(System.in);

System.out.print("Enter phone number (3 digit areacode + 7 digit number) :");

return "+1" + input.nextLine().replaceAll("[^0-9]","");

}

}

**class** FrStudent extends Student {

FrStudent() {

this.setNationality("French");

}

@Override

public String acceptName() {

Scanner input = new Scanner(System.in);

System.out.print("Enter first name: ");

String firstName = input.nextLine().replaceAll("[^a-zA-Z]","");

System.out.print("Enter family name: ");

String familyName = input.nextLine().replaceAll("[^a-zA-Z]","");

return firstName + " " + familyName;

}

@Override

public String acceptPhoneNumber() {

Scanner input = new Scanner(System.in);

System.out.print("Enter phone number (1 digit areacode + 7 digit number): ");

return "+33" + input.nextLine().replaceAll("[^0-9]","");

}

}

**NOTES**

* str.replaceAll("[^a-zA-Z]","") removes all non-letters of the string str
* str.replaceAll("[^0-9]","") removes all non-digits of the string str