Assignment 3

/*

Deadline： **Feb 1, 23:59 PST**

Please submit **.java files**

Extra credits are added only if total score is less than 10

If you have any questions, the TA office hour is on **Tuesday**, from **1 pm -- 4 pm**, in **401 - 142**.

*/

1. (score: 5) Design a simple registration system that allows Student to register in a course using 2 classes: class Student & class Course. Implement the scenarios in class Test's main method.

Each student has a name and an id variable. Each object of class Student is initialized using values of name and id passed to constructor. Class Student has accessor methods for its instance variables.

Each Course has a name, and a variable numberOfStudent representing the number of registered students. A course can have a maximum number of 10 students registered in it. Class Course store the registered students in students which is an array of type Student.

When a student register in a course, he is added to the array. Each object of class Course is initialized using the title.

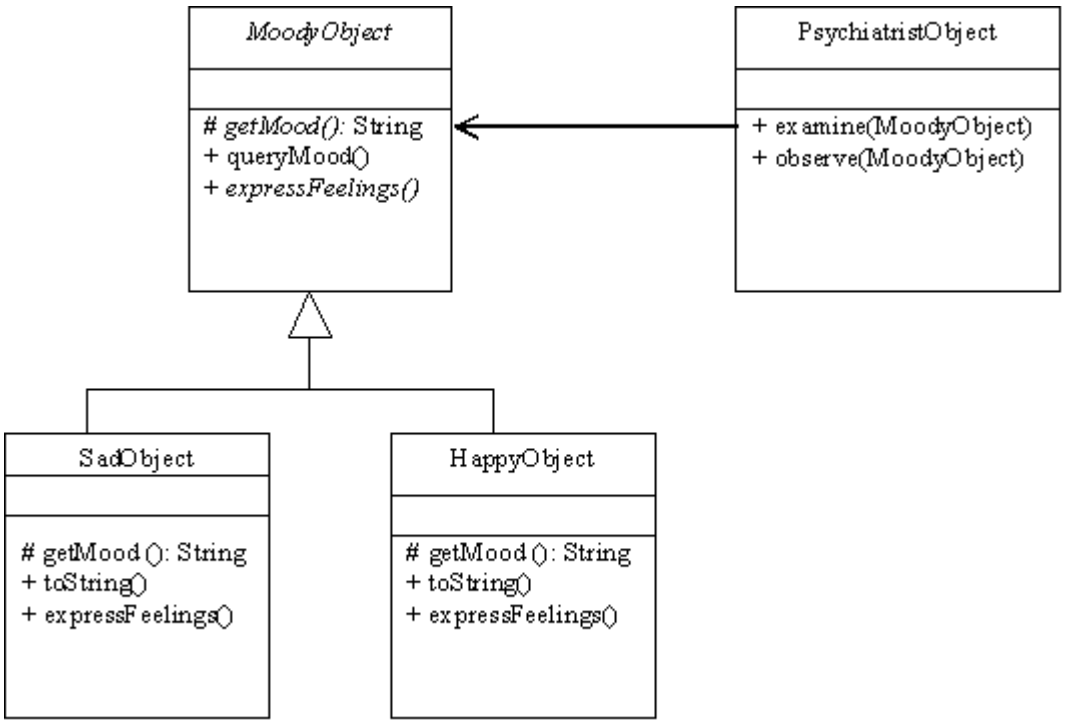
Class Course has the following methods:

method getStudents(): return the array of registered students;

method boolean isFull(): return true if the course is full, accessor method for the title and numberOfStudent field;

method registerStudent (Student student): if the course is not full, register a student in course.

2. (score: 5) For this problem you will write a Java program that uses polymorphism and abstract classes and methods. The program should implement the design indicated in this UML diagram



As indicated in the diagram, each class has the following methods:

MoodyObject:

```
//returns the mood: sad or happy - depending on which object sends the message
protected abstract String getMood();

//each object expresses a different emotion
protected abstract void expressFeelings();

//an object responds according to how it feels, print "I feel happy(or sad) today!"
public void queryMood() {
```

SadObject:

```
//returns a string indicating sad
protected String getMood();

//print crying string: " 'wah' 'boo hoo' 'weep' 'sob' 'weep' "
public void expressFeelings();

//returns message about self: "Subject cries a lot";
public String toString();
```

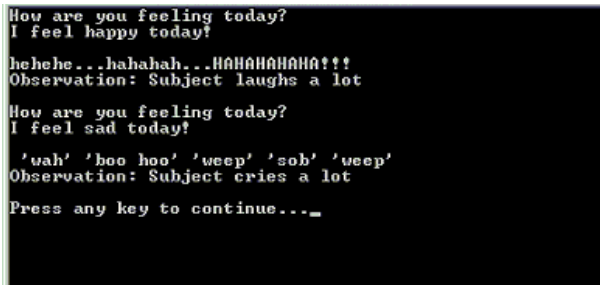
HappyObject:

```
//returns a string indicating happy
protected String getMood();

//print laughter string: "hehehe...hahahah...HAHAHAHAHA!!!"
public void expressFeelings();

//returns message about self: "Subject laughs a lot"
public String toString();
```

PsychiatristObject:

```
//asks a moody object about its mood
public void examine(MoodyObject moodyObject);

//a moodyObject is observed to either laugh or cry
public void observe(MoodyObject moodyObject);
```

Write a main() method that creates a psychiatrist object and two moodyObjects. The psychiatrist object will examine and observe each moodyObject. The output of your program should be same with the following output:



3. (score: 2) Extra credit:

/*

Given an input string, reverse the string word by word. Example: Input: “The sky is blue”, output: “blue is sky the”.

1. A word is defined as a sequence of non-space characters.

2. Input string may contain leading or trailing spaces. However, your reversed string should **not** contain leading or trailing spaces.

3.You need to reduce multiple spaces between two words to a single space in the reversed string.

*/

```
public String reverse(String s) {


}
```