import java.util.*;

public class Adventure {
/* Version 1
   This program is an arithmetic adventure game where an adventurer navigates rooms
   that contain treasure chests that are opened by correctly answering arithmetic
   problems.
*/
    public static void main(String args[]) {
    /* Program statements go here. */
    String name;
    Integer tokens;

    System.out.println("Welcome to the Arithmetic Adventure game.");
    System.out.print("The date is ");
    System.out.println(new Date());
    System.out.println();
    System.out.print("What is your name?");
    name = Keyboard.in.readString();
    System.out.print("Well ");
    System.out.print(name);
    System.out.println(", after a day of hiking you spot a silver cube.");
    System.out.println("The cube appears to be about 5 meters on each side.");
    System.out.println("You find a green door, open it and enter.");
    System.out.println("The door closes behind you with a soft whir and
disappears.");
    System.out.println("There is a feel of mathematical magic in the air.");

    Keyboard.in.pause();
    System.out.print("How many tokens would you like?");
    tokens = Keyboard.in.readInteger();
    System.out.print("Congratulations ");
    System.out.print(name);
    System.out.println(" you have left the game with ");
    System.out.println(tokens);
    System.out.println(" tokens.");
    }
}
import java.util.*;

public class Adventure {

    /* Version 2
        This program is an arithmetic adventure game ... */

    /* Constructors */
    public Adventure () {
        /* Initialize an Adventure by creating the appropriate objects. */
    }

    /* Main program */
    public static void main(String args[]) {
        Adventure game;
        game = new Adventure();
        game.play();
    }

    /* Private Instance Methods */
    private void play() {
        /* Play the Adventure game. */
        String name;
        Integer tokens;
        name = this.greeting();
        tokens = this.enterRoom(name);
        this.farewell(name, tokens);
    }

    private void farewell(String userName, Integer tokenCount) {
        /* Say farewell to the user with the given name and report the given count of tokens earned. */
        System.out.print("Congratulations ");
        System.out.print(userName);
        System.out.print(" you have left the game with ");
        System.out.print(tokenCount);
        System.out.println(" tokens.");
    }

    private String greeting() {
        /*
            Greeting of the user.
        */
    }

    private Integer enterRoom(String name) {
        /*
            Enter room with the given name.
        */
    }

    private void farewell() {
        /*
            farewell method.
        */
    }
}
Greet the user and answer a String that represents the player’s name.

*/
String playerName;

System.out.println("Welcome to the Arithmetic Adventure game.");
System.out.print("The date is ");
System.out.println(new Date());
System.out.println();
System.out.print("What is your name?");
playerName = Keyboard.in.readString();
System.out.print("Well ");
System.out.println(playerName);
System.out.println("", after a day of hiking you spot a silver cube.");
System.out.println("The cube appears to be about 5 meters on each side.");
System.out.println("You find a green door, open it and enter.");
System.out.println("The door closes behind you with a soft whir and disappears.");
System.out.println("There is a feel of mathematical magic in the air.");
Keyboard.in.pause();
return playerName;
}

private Integer enterRoom(String theName) {

/*
The user with the given name has entered the first room. After the adventure is done, return the number of tokens obtained during the game.
*/
Integer tokenCount;

System.out.print("How many tokens would you like, ");
System.out.print(theName);
System.out.print("?");
tokenCount = Keyboard.in.readInteger();
return tokenCount;
}
import java.util.*;

public class Adventure {

/* Version 3

This program is an arithmetic adventure game where an
adventurer navigates rooms that contain treasure chests
that are opened by correctly answering arithmetic problems.
*/

/* Constructors */
public Adventure () {
    /* Initialize an Adventure by creating the appropriate
     * objects.
    */
}

/* Main program */
public static void main(String args[]) {
    Adventure game;
    game = new Adventure();
    game.play();
}

/* Private Instance Methods */
private void play() {
    /* Play the Adventure game. */
    Adventurer adventurer;
    adventurer = this.greeting();
    this.enterRoom(adventurer);
    this.farewell(adventurer);
}

private void farewell(Adventurer adventurer) {
    /* Say farewell to the user and report the game result. */
    System.out.print("Congratulations ");
    System.out.print(adventurer.name());
    System.out.print(" you have left the game with ");
    System.out.print(adventurer.tokens());
    System.out.println(" tokens.");
}

private Adventurer greeting() {
    /* Greet the user and answer an Adventurer that represents
     * the user. */
    String playerName;
    System.out.println("Welcome to the Arithmetic Adventure game.");
    System.out.print("The date is ");
}
System.out.println(new Date());
System.out.println();
System.out.println("What is your name?");
playerName = Keyboard.in.readString();
System.out.println("Well ");
System.out.println(playerName);
System.out.println("", after a day of hiking you spot a silver cube.");
System.out.println("The cube appears to be about 5 meters on each side.");
System.out.println("You find a green door, open it and enter.");
System.out.println("The door closes behind you with a soft whir and disappears.");
System.out.println("There is a feel of mathematical magic in the air.");
Keyboard.in.pause();
return new Adventurer(playerName);
}

private void enterRoom(Adventurer adventurer) {
    /*
    The given adventurer has entered the first room.
    */
    Integer myTokens;

    System.out.print("How many tokens would you like, ");
    System.out.print(adventurer.name());
    System.out.print("?");
    myTokens = Keyboard.in.readInt();
    adventurer.gainTokens(myTokens.intValue());
}

public class Adventurer {
    /*
    An instance of this class represents a player of the Adventure game.
    */

    /* Constructors */
    public Adventurer(String nameString) {
        /*
        Initialize me with the given name and zero tokens.
        */
        this.name = nameString;
        this.tokens = 0;
    }

    /* Instance Methods */

    public String name() {
        /*
        Answer a String representing my name.
        */
        return this.name;
    }

    public int tokens() {
        /*
        Answer my number of Tokens.
        */
        return this.tokens;
    }
public void gainTokens(int anInt) {
    /*
     * Add the given number of tokens to my total.
     */
    this.tokens = this.tokens + anInt;
}

public void loseTokens(int anInt) {
    /*
     * Remove the given number of tokens from my total.
     */
    this.tokens = this.tokens - anInt;
}

    public void reportTokens() {
    /*
     * Output the number of tokens I have.
     */
    System.out.print("You have ");
    System.out.print(this.tokens);
    System.out.println(" tokens in your pocket.");
    }

    /* Private Instance Variables */

    private String name;
    private int tokens;
}
1. We are going to add some functionality to the Arithmetic Adventure game.
2. We will put treasure chests in rooms.
3. When the adventurer tries to open a chest we will generate an arithmetic question.
4. The chest will contain a random number of tokens that will be added or subtracted to the adventurer's total depending on whether the adventurer answers the question correctly.

In the Adventure class we will:
- replace the method enterRoom(Adventurer)
- Add a class called Chest.
- Add a class called Question.
- Add a class called RandomInt
- Leave the Adventurer class unchanged. See Adventure Code Version 3 for this code.

import java.util.*;
public class Adventure {
   /* Version 4
        This program is an arithmetic adventure game …
   */

   /* Constructors */
   public Adventure () {
      /* Initialize an adventure by creating the appropriate objects.
      */
   }

   /* Main program */
   public static void main(String args[]) {
      Adventure game;
      game = new Adventure();
      game.play();
   }

   /* Private Instance Methods */
   private void play() {
      /* Plays the Adventure game.
      */
      Adventurer adventurer;
      adventurer = this.greeting();
      this.enterRoom(adventurer);
      this.farewell(adventurer);
   }

   private Adventurer greeting() {
      /* Great the user and answer an Adventurer that represents the user.
      */
   }
}
String playerName;

System.out.println("Welcome to the Arithmetic Adventure game.");
System.out.print("The date is ");
System.out.println(new Date());
System.out.println();
System.out.println("What is your name?");
playerName = Keyboard.in.readString();
System.out.print("Well ");
System.out.print(playerName);
System.out.println("", after a day of hiking you spot a silver cube.");
System.out.println("The cube appears to be about 5 meters on each side.");
System.out.println("You find a green door, open it and enter.");
System.out.println("The door closes behind you with a soft whir and disappears.");
System.out.println("There is a feel of mathematical magic in the air.");
Keyboard.in.pause();
return new Adventurer(playerName);

private void enterRoom(Adventurer adventurer) {
    /*
    The given adventurer has entered the first room.
    */
    Chest chest = new Chest();
    chest.display();
    chest.open(adventurer);
}

private void farewell(Adventurer adventurer) {
    /*
    Say farewell to the user and report the game result.
    */
    System.out.print(" Congratulations ");
    System.out.print(adventurer.name());
    System.out.println(" you have left the game with ");
    System.out.print(adventurer.tokens());
    System.out.println(" tokens.");
}

import java.util.*;
public class Chest {
    /*
    An instance of this class represents a treasure chest in the Adventure game. A Chest contains a number of tokens.
    */
    public Chest() {
        /*
        Initialize me so that I contain a random number of tokens.
        */
        this.tokens = Chest.generator.next(Chest.maxTokens);
    }
}
/* Instance Methods */

public void display() {
    /* Output a description of myself. */
    System.out.println("There is a small carved chest in the center of the room.");
    System.out.println("It appears to be a treasure chest!");
}

public void open(Adventurer adventurer) {
    /* Ask the user an arithmetic question and if a correct answer is given, add tokens to the given Adventurer. If it is answered incorrectly, remove tokens. */
    Question question;
    question = new Question();
    question.ask();
    // We really want to do only one of the next two lines, depending on the user’s answer.
    this.correctAnswer(adventurer);
    this.wrongAnswer(question, adventurer);
}

/* Private Static Variables */

private static final int maxTokens = 10;
private static final RandomInt generator = new RandomInt(1);

/* Private Instance Variables */

private int tokens;

/* Private Instance Methods */

private void correctAnswer(Adventurer adventurer) {
    /* Congratulate the adventurer and add some tokens. */
    System.out.println();
    System.out.println("A small loudspeaker appears in the air.");
    System.out.println("You hear the sound of a harp and a pleasant voice says congratulations.");
    System.out.print("The lid of the chest opens to reveal ");
    System.out.print(this.tokens);
    System.out.println(" valuable tokens.");
    System.out.println("They literally fly into your pocket and the chest disappears.");
    adventurer.gainTokens(this.tokens);
    adventurer.reportTokens();
}

private void wrongAnswer(Question question, Adventurer adventurer) {
    /* Report the correct answer and remove some tokens from the given adventurer. */
    int loss;
    System.out.println();
    System.out.println("A small loudspeaker appears in the air.");
    System.out.println("You hear the sound of a deep gong and a pleasant voice says:");
    System.out.print("Sorry, the correct answer is ");
    System.out.println(question.answer());
    System.out.println(".");
}
loss = Math.min(this.tokens, adventurer.tokens());
System.out.print("You see ");
System.out.print(loss);
System.out.println(" valuable tokens fly out of your pocket and fall to the floor.");
System.out.println("A small vacuum cleaner appears, sweeps up your scattered tokens and disappears.");
adventurer.loseTokens(loss);
adventurer.reportTokens();

import java.util.*;
public class Question {

    /* Constructor */
    public Question() {
        this.leftOperand = Question.generator.next(Question.maxOperand);
        this.rightOperand = Question.generator.next(Question.maxOperand);
    }

    /* Instance Methods */
    public void ask() {
        Integer answer;
        System.out.print(this.leftOperand);
        System.out.print(" + ");
        System.out.print(this.rightOperand);
        System.out.print(" = ");
        answer = Keyboard.in.readInteger();
    }

    public int answer() {
        return this.leftOperand + this.rightOperand;
    }

    /* Private Static Variables */
    private static final int maxOperand = 9;
    private static final RandomInt generator = new RandomInt(2);

    /* Private Instance Variables */
    private int leftOperand;
    private int rightOperand;
}

import java.util.*;
public class RandomInt {
    /*
     * An instance of this class represents a generator that can generate a series of random positive ints.
     */

    /* Constructor */
    public RandomInt(int seed) {
        /*
        * Initialize me so that I use the given seed.
        */
        this.generator = new Random(seed);
    }

    /* Instance Methods */
    public int next(int max) {
        /*
        * Answer a Random int between 1 and the given max.
        */
        return Math.round(max * this.generator.nextFloat() - 0.5f) + 1;
    }

    /* Private Instance Variables */
    private Random generator;
}
**Adventure Code Version 5**

1. We are going to add some functionality to the Arithmetic Adventure game.
2. We will change the ask() method in class Question so it checks the user's answer against the correct answer and returns a boolean value true or false.
   - We will change the open() method in the class Chest so that if the ask() message returns true then we will gain tokens and if returns false then we will remove tokens.
4. Leave the Adventure class unchanged except for changing the comment to Version 5.
   - Leave the Adventurer class unchanged.
   - Leave the RandomInt class unchanged.
   - See Versions 3 for the Adventurer class.
   - See Version 4 for the Adventure class and the RandomInt class.

```java
import java.util.*;
public class Chest {

    /* An instance of this class represents a treasure chest in the Adventure game. A Chest contains a number of tokens. */

    /* Constructor */
    public Chest() {
        /* Initialize me so that I contain a random number of tokens. */
        this.tokens = Chest.generator.next(Chest.maxTokens);
    }

    /* Instance Methods */
    public void display() {
        /* Output a description of myself. */
        System.out.println("There is a small carved chest in the center of the room.");
        System.out.println("It appears to be a treasure chest!");
    }

    public void open(Adventurer adventurer) {
        /* Ask the user an arithmetic question and if a correct answer is given, add tokens to the given Adventurer. If it is answered incorrectly, remove tokens. */
        Question question;
        question = new Question();
        if (question.ask())
            this.correctAnswer(adventurer);
        else
            this.wrongAnswer(question, adventurer);
    }

    /* Private Static Variables */
    private static final int maxTokens = 10;
    private static final RandomInt generator = new RandomInt(1);
```
/* Private Instance Variables */

private int tokens;

/* Private Instance Methods */

private void correctAnswer(Adventurer adventurer) {
    /* Congratulate the adventurer and add some tokens. */
    System.out.println();
    System.out.println("A small loudspeaker appears in the air.");
    System.out.println("You hear the sound of a harp and a pleasant voice says congratulations.");
    System.out.print("The lid of the chest opens to reveal ");
    System.out.print(this.tokens);
    System.out.println(" valuable tokens.");
    System.out.println("They literally fly into your pocket and the chest disappears.");
    adventurer.gainTokens(this.tokens);
    adventurer.reportTokens();
}

private void wrongAnswer(Question question, Adventurer adventurer) {
    /* Report the correct answer and remove some tokens from the given adventurer. */
    int loss;

    System.out.println();
    System.out.println("A small loudspeaker appears in the air.");
    System.out.println("You hear the sound of a deep gong and a pleasant voice says:");
    System.out.println("Sorry, the correct answer is ");
    System.out.println(question.answer());
    System.out.println(".");

    loss = Math.min(this.tokens, adventurer.tokens());
    System.out.println("You see ");
    System.out.println(loss);
    System.out.println(" valuable tokens fly out of your pocket and fall to the floor.");
    System.out.println("A small vacuum cleaner appears, sweeps up your scattered tokens and disappears.");
    adventurer.loseTokens(loss);
    adventurer.reportTokens();
}
import java.util.*;
public class Question {

    /*
     * An instance of this class represents an arithmetic problem in the Arithmetic
     * Adventure game.
     */

    /* Constructor */
    public Question() {
        /*
         * Initialize me so that I have two operands.
         */
        this.leftOperand = Question.generator.next(Question.maxOperand);
        this.rightOperand = Question.generator.next(Question.maxOperand);
    }

    /* Instance Methods */
    public boolean ask() {
        /*
         * Pose myself. Return true if the user’s response
         * was correct and false otherwise.
         */
        Integer answer;
        System.out.print(this.leftOperand);
        System.out.print(" + ");
        System.out.print(this.rightOperand);
        System.out.print(" = ");
        answer = Keyboard.in.readInteger();
        return answer.intValue() == this.answer();
    }

    public int answer() {
        /*
         * Answer my correct answer.
         */
        return this.leftOperand + this.rightOperand;
    }

    /* Private Static Variables */
    private static final int maxOperand = 9;
    private static final RandomInt generator = new RandomInt(2);

    /* Private Instance Variables */
    private int leftOperand;
    private int rightOperand;
}
**Adventure Code Version 6**

1. Time to add more functionality to the Arithmetic Adventure game.
2. We create a room for the Adventurer and allow the user to either open the chest in the room or quit.
3. To do this we will create two new classes, Room and TextMenu.
4. Later we will add more rooms, add doors to the room and allow the user to open a door and enter another room.
5. In the Adventure class we will:
   - Modify the constructor Adventure() by adding code.
   - Add an an instance variable that is bound to the first Room object that the user enters.
   - Modify the play() method.
6. Leave the classes: Question, Chest, RandomInt and Adventurer unchanged

```java
import java.util.*;
public class Adventure {
    /* Version 6
     * This program is an arithmetic adventure game ...
     */

    /* Constructors */
    public Adventure () {
        /* Initialize an adventure by creating the appropriate objects. */
        this.firstRoom = new Room(1);
    }

    /* Main program */
    public static void main(String args[]) {
        Adventure game;
        game = new Adventure();
        game.play();
    }

    /* Private Instance Variables */
    private Room firstRoom;

    /* Private Instance Methods */
    private void play() {
        /* Plays the Adventure game. */
        Adventurer adventurer;
        Room room;
        adventurer = this.greeting();
        room = firstRoom.enter(adventurer);
        this.farewell(adventurer);
    }
```
private Adventurer greeting() {
    /*
     * Great the user and answer an Adventurer that
     * represents the user.
     */
    String playerName;

    System.out.println("Welcome to the Arithmetic Adventure game.");
    System.out.print("The date is ");
    System.out.println(new Date());
    System.out.println();
    System.out.print("What is your name?");
    playerName = Keyboard.in.readString();

    System.out.print("Well ");
    System.out.print(playerName);
    System.out.println("", after a day of hiking you spot a silver cube.");
    System.out.println("The cube appears to be about 5 meters on each side.");
    System.out.println("You find a green door, open it and enter.");
    System.out.println("The door closes behind you with a soft whir and disappears.");
    System.out.println("There is a feel of mathematical magic in the air.");
    Keyboard.in.pause();
    return new Adventurer(playerName);
}
private void farewell(Adventurer adventurer) {
    /*
     * Say farewell to the user and report the game result.
     */
    System.out.print("Congratulations ");
    System.out.print(adventurer.name());
    System.out.print(" you have left the game with ");
    System.out.print(adventurer.tokens());
    System.out.println(" tokens.");
}

import java.util.*;
public class Room {
    /*
     * A room contains a treasure chest and some doors to adjoining rooms.
     */
    /* Constructor */
    public Room(int anInt) {
        /*
         * Initialize me so that I have the given room number,
         * contain a treasure chest, and no doors.
         */
        this.number = anInt;
        this.chest = new Chest();
    }
    /* Instance Methods */
    public Room enter(Adventurer adventurer) {
        /*
         * Describe myself, display a list of options, and
         * perform the selected option. If the user selected
         * quit then return null. If the user selected to go
to another Room then return that Room. Otherwise return this Room.

*/

TextMenu menu;
String action;
this.display();
menu = this.buildMenu();
action = menu.launch();
return this.performAction(action, adventurer);

}*/

/* Private Instance Variables */
private Chest chest;
private int number;

/* Private Instance Methods */

private void display() {
    /* Output a description of myself. */
    this.displayBasic();
    this.displayDoors();
    if (this.chest != null)
        this.chest.display();
}

private void displayBasic() {
    /* Output a basic description of myself that is independent of my doors and contents. */
    System.out.println();
    System.out.println("You are in a cubic room, 5 meters on each side.");
    System.out.println("A soft yellow glow illuminates the room.");
    System.out.println("The walls are made of a silver metal.");
    System.out.println("There is a large number ");
    System.out.println(this.number);
    System.out.println(" painted on one wall.");
}

private void displayDoors() {
    /* Output a description of all of my doors. */
}

private TextMenu buildMenu() {
    /* Create and answer a TextMenu containing the user's valid actions. */
    TextMenu menu;

    menu = new TextMenu();
    menu.add("Quit");
    if (this.chest != null)
        menu.add("Open the chest.");
    // Add door choices here
    return menu;
}

private Room performAction(String action, Adventurer adventurer) {
    /* Perform the action described by the given String for
the given Adventurer. Return the room the user
selected, null if the user selected quit and this
room if the user selected to open the chest.
 */
if (action.equals("Open the chest.")) {
    this.chest.open(adventurer);
    this.chest = null;
    return this;
}
if (action.equals("Quit"))
    return null;
return null;

import java.io.*;
import java.util.*;
public class TextMenu {

    /* An instance of this class displays a list of strings for the user and allows
    the user to pick one. For now, up to five entries are supported. */
    /* Constructor */
    public TextMenu() {
	/*
        Initialize me with no entries.
	*/
    }
    /* Instance Methods */
    public void add(String entry) {
        /*
            Add the given String to me as my next choice.
        */
        if (entry1 == null) {
            this.entry1 = entry;
            return;
        }
        if (entry2 == null) {
            this.entry2 = entry;
            return;
        }
        //more of the same for entries 3, 4 and 5.
    }
    public String launch() {
        /*
            Display myself and answer the String entry selected by the user.
        */
        Integer choice;
        int index;
        this.display();
        choice = Keyboard.in.readInteger();
        if (choice == null)
            return this.entry1;
        index = choice.intValue();
        switch (index) {
            case 1: return this.entry1;
            case 2: return this.entry2;
            case 3: return this.entry3;
            case 4: return this.entry4;
            case 5: return this.entry5;
        }
        return null;
    }
}
case 5: return this.entry5;
    default: return this.entry1;
}

/* Private Instance Variables */
private String entry1;
private String entry2;
private String entry3;
private String entry4;
private String entry5;

/* Private Instance Methods */
private void display() {
    /*
    Display myself on the screen.
    */
    String entry;
    int index;
    System.out.println();
    System.out.println("Please type a number and press the Enter key:");
    if (this.entry1 != null) {
        System.out.print("1. ");
        System.out.println(this.entry1);
    }
    if (this.entry2 != null) {
        System.out.print("2. ");
        System.out.println(this.entry2);
    }
    // same code for entry2, entry3, entry4 and entry5
    }
}
Adventure Code Version 7

1. In the TextMenu class we will:
   - Add an instance variable called size which indicates how many legal entries I have.
   - Modify the constructor TextMenu().
   - Modify the instance method add().
   - Replace instance method launch().
   - Add instance method getUserSelection().

2. In the Question class we will:
   - Replace the ask() method.
   - Add a display() method.

3. Leave the classes: Adventure, Adventurer, RandomInt, Chest and Room unchanged.

```java
import java.io.*;
import java.util.*;
public class TextMenu {
    /**
     * An instance of this class displays a list of strings for the user and allows
     * the user to pick one. For now, up to five entries are supported.
     */
    public TextMenu() {
        /* Initialize me with no entries. */
        this.size = 0;
    }

    /* Instance Methods */
    public void add(String entry) {
        /* Add the given String to me as my next choice. */
        this.size = this.size + 1;
        if (entry1 == null) {
            this.entry1 = entry;
            return;
        }
        if (entry2 == null) {
            this.entry2 = entry;
            return;
        }
        //more of the same for entries 3, 4 and 5.
    }

    public String launch() {
        /* Display myself and answer the String entry selected
         * by the user. */
        String action;
        int index;

        index = this.getUserSelection();
        switch (index) {
            case 1: action = this.entry1; break;
            //more cases for entries 2, 3, 4 and 5.
        }
    }
}
```
case 2: action = this.entry2; break;
case 3: action = this.entry3; break;
case 4: action = this.entry4; break;
case 5: action = this.entry5; break;
default: action = "";
}
return action;

/* Private Instance Variables */
private String entry1;
private String entry2;
private String entry3;
private String entry4;
private String entry5;
private int size;

/* Private Instance Methods */
private void display() {
/*
Display myself on the screen.
*/
String entry;
int index;
System.out.println();
System.out.println("Please type a number and press the Enter key:");
if (this.entry1 != null) {
    System.out.print("1. ");
    System.out.println(this.entry1);
}
if (this.entry2 != null) {
    System.out.print("2. ");
    System.out.println(this.entry2);
}
// same code for entry3, entry4 and entry5
}
private int getUserSelection() {
/*
Query the user for an action and answer the index of
the choice. If the user does not answer with a valid
action, query again.
*/
Integer choice;
int index;

index = 0;
while ([(index < 1) || (index > this.size)] {
    this.display();
    choice = Keyboard.in.readInteger();
    if (choice == null)
        index = 0;
    else
        index = choice.intValue();
}
return index;
}
import java.util.*;
public class Question {
    /*
     An instance of this class represents an arithmetic problem in the Arithmetic
     Adventure game.
     */

    /* Constructor */
    public Question() {
        /* Initialize me so that I have two operands. */
        this.leftOperand = Question.generator.next(Question.maxOperand);
        this.rightOperand = Question.generator.next(Question.maxOperand);
    }

    /* Instance Methods */
    public boolean ask() {
        /* Pose myself. Return true if the user’s response
        was correct and false otherwise. */
        Integer answer;
        answer = null;
        while (answer == null) {
            this.display();
            answer = Keyboard.in.readInteger();
        }
        return answer.intValue() == this.answer();
    }

    public int answer() {
        /* Answer my correct answer. */
        return this.leftOperand + this.rightOperand;
    }

    /* Private Static Variables */
    private static final int maxOperand = 9;
    private static final RandomInt
    generator = new RandomInt(2);

    /* Private Instance Variables */
    private int leftOperand;
    private int rightOperand;

    /* Private Instance Methods */
    private void display() {
        /* Display myself. */
        System.out.print(this.leftOperand);
        System.out.print(" + ");
        System.out.print(this.rightOperand);
        System.out.print(" = ");
    }
}
import java.util.*;  
public class Adventure {  
/* Version 8  
This program is an arithmetic adventure game ...  
*/  
/* Constructors */  

public Adventure () {  
/* Initialize an adventure by creating the appropriate objects. */  
    Vector rooms;  
    int i;  
    rooms = new Vector();  
    for (i = 0; i <= 4; i++)  
        rooms.addElement(new Room(i + 1));  
    this.makeDoor(rooms, 1, 2, "red");  
    this.makeDoor(rooms, 1, 3, "blue");  
    this.makeDoor(rooms, 2, 4, "green");  
    this.makeDoor(rooms, 2, 5, "blue");  
    this.firstRoom = (Room) rooms.elementAt(0);  
}  
/* Main program */  

public static void main(String args[]) {  
    Adventure game;  
    game = new Adventure();  
    game.play();  
    }  
/* Private Instance Variables */  

private Room firstRoom;  
/* Private Instance Methods */  

private void play() {  
/* Plays the Adventure game. */  
    Adventurer adventurer;  
    Room room;  
}
adventurer = this.greeting();
room = firstRoom;
while (room != null)
    room = room.enter(adventurer);
this.farewell(adventurer);
}

private Adventurer greeting() {
    /*
     * Great the user and answer an Adventurer that represents the user.
     */
    String playerName;

    System.out.println("Welcome to the Arithmetic Adventure game.");
    System.out.println("The date is ");
    System.out.println(new Date());
    System.out.println();
    System.out.print("What is your name?");
    playerName = Keyboard.in.readString();
    System.out.print("Well ");
    System.out.println(playerName);
    System.out.println(", after a day of hiking you spot a silver cube.");
    System.out.println("The cube appears to be about 5 meters on each side.");
    System.out.println("You find a green door, open it and enter.");
    System.out.println("The door closes behind you with a soft whirl and disappears.");
    System.out.println("There is a feel of mathematical magic in the air.");
    Keyboard.in.pause();
    return new Adventurer(playerName);
}

private void farewell(Adventurer adventurer) {
    /*
     * Say farewell to the user and report the game result.
     */
    System.out.print("Congratulations ");
    System.out.print(adventurer.name());
    System.out.print(" you have left the game with ");
    System.out.print(adventurer.tokens());
    System.out.println(" tokens.");
}

private void makeDoor(Vector myRooms, int from, int to, String color) {
    /*
     * Make a Door from the Room with the given room number to the Room with the given room number in the given Vector of rooms. Use the given Door color.
     */
    Room fromRoom;
    Room toRoom;

    fromRoom = (Room) myRooms.elementAt(from - 1);
    toRoom = (Room) myRooms.elementAt(to - 1);
    fromRoom.makeDoor(toRoom, color);
}
public class Adventurer {

    /*
    An instance of this class represents a player of the Adventure game.
    */

    /* Constructors */
    public Adventurer(String name) {
        /*
        Initialize me with the given name and zero tokens.
        */
        this.name = name;
        this.tokens = 0;
    }

    /* Instance Methods */

    public String name() {
        /*
        Answer a String representing my name.
        */
        return this.name;
    }

    public int tokens() {
        /*
        Answer my number of Tokens.
        */
        return this.tokens;
    }

    public void gainTokens(int anInt) {
        /*
        Add the given number of tokens to my total.
        */
        this.tokens = this.tokens + anInt;
    }

    public void loseTokens(int anInt) {
        /*
        Remove the given number of tokens from my total.
        */
        this.tokens = this.tokens - anInt;
    }

    public void reportTokens() {
        /*
        Output the number of tokens I have.
        */
        System.out.print("You have ");
        System.out.print(this.tokens);
        System.out.println(" tokens in your pocket.");
    }

    /* Private Instance Variables */

    private String name;
    private int tokens;
}

import java.util.*;
public class Room {

    /*
     * A room contains a treasure chest and some doors to adjoining rooms.
     */

    /* Constructor */
    public Room(int anInt) {
        /*
         * Initialize me so that I have the given room number,
         * contain a treasure chest, and no doors.
         */
        this.number = anInt;
        this.chest = new Chest();
        this.doors = new Vector();
    }

    /* Instance Methods */
    public Room enter(Adventurer adventurer) {
        /*
         * Describe myself, display a list of options, and perform the selected
         * option. If the user selected quit then return null. If the user selected
         * to go to another Room then return that Room. Otherwise return this Room.
         */
        TextMenu menu;
        String action;
        this.display();
        menu = this.buildMenu();
        action = menu.launch();
        return this.performAction(action, adventurer);
    }

    public void makeDoor(Room aRoom, String color) {
        /*
         * Make a door of the given color and place it between
         * me and the given Room.
         */
        Door door;
        door = new Door(color, this, aRoom);
        this.doors.addElement(door);
        aRoom.doors.addElement(door);
    }

    /* Private Instance Variables */
    private Chest chest;
    private int number;
    private Vector doors;

    /* Private Instance Methods */
    private void display() {
        /*
         * Output a description of myself.
         */
        this.displayBasic();
        this.displayDoors();
        if (this.chest != null)
            chest.display();
    }
}
private void displayBasic() {
/*
   Output a basic description of myself that is
   independent of my doors and contents.
*/
    System.out.println();
    System.out.println("You are in a cubic room, 5 meters on each side.");
    System.out.println("A soft yellow glow illuminates the room.");
    System.out.println("The walls are made of a silver metal.");
    System.out.print("There is a large number ");
    System.out.println(this.number);
    System.out.println(" painted on one wall.");
}
private void displayDoors() {
/*
   Output a description of all of my doors.
*/
    Door door;
    int index;

    for (index = 0; index < this.doors.size(); index++){
        door = (Door) this.doors.elementAt(index);
        door.display();
    }
}
private TextMenu buildMenu() {
/*
   Create and answer a TextMenu containing the user's valid actions.
*/
    TextMenu menu;
    int index;
    Door door;

    menu = new TextMenu();
    menu.add("Quit");
    if (this.chest != null)
        menu.add("Open the chest.");
    for (index = 0; index < this.doors.size(); index++){
        door = (Door) this.doors.elementAt(index);
        menu.add("Open the " + door.color() + " door.");
    }
    return menu;
}
private Room performAction(String action, Adventurer adventurer) {
/*
   Perform the action described by the given String for the given
   Adventurer. Return the room the user selected, null if the user selected
   quit and this room if the user selected to open the chest.
*/
    if (action.equals("Open the chest.")) {
        this.chest.open(adventurer);
        this.chest = null;
        return this;
    }
    if (action.equals("Quit"))
        return null;
    return this.getRoomForAction(action);
}
private Room getRoomForAction(String action) {
    /*
     * Return the Room that I am connected to that is
     * represented by the given action String. If no such
     * Door exists, return me.
     */
    int index;
    String color;
    Door door;

    color = action.substring(9, action.length() - 6);
    for (index = 0; index < this.doors.size(); index++) {
        door = (Door) this.doors.elementAt(index);
        if (color.equals(door.color()))
            return door.adjoiningRoom(this);
    }
    return this;
}

import java.util.*;
public class Chest {
    /*
     * An instance of this class represents a treasure chest in
     * the Adventure game. A Chest contains a number of tokens.
     */
    /* Constructor */
    public Chest() {
        /*
         * Initialize me so that I contain a random number of
         * tokens.
         */
        this.tokens = Chest.generator.next(Chest.maxTokens);
    }

    /* Instance Methods */
    public void display() {
        /*
         * Output a description of myself.
         */
        System.out.println("There is a small carved chest in the center of the
room.");
        System.out.println("It appears to be a treasure chest!");
    }

    public void open(Adventurer adventurer) {
        /*
         * Ask the user an arithmetic question and if a correct
         * answer is given, add tokens to the given Adventurer. If it is
         * answered incorrectly, remove tokens.
         */
        Question question;
        question = new Question();
        if (question.ask())
            this.correctAnswer(adventurer);
        else
            this.wrongAnswer(question, adventurer);
    }
}
/* Private Static Variables */

    private static final int maxTokens = 10;
    private static final RandomInt generator = new RandomInt(1);

/* Private Instance Variables */

    private int tokens;

/* Private Instance Methods */

    private void correctAnswer(Adventurer adventurer) {
        /* Congratulate the adventurer and add some tokens.*/
        System.out.println();
        System.out.println("A small loudspeaker appears in the air.");
        System.out.println("You hear the sound of a harp and a pleasant voice says congratulations.");
        System.out.print("The lid of the chest opens to reveal ");
        System.out.print(this.tokens);
        System.out.println(" valuable tokens.");
        System.out.println("They literally fly into your pocket and the chest disappears.");
        adventurer.gainTokens(this.tokens);
        adventurer.reportTokens();
    }

    private void wrongAnswer(Question question, Adventurer adventurer) {
        /* Report the correct answer and remove some tokens from the given adventurer. */
        int loss;
        System.out.println();
        System.out.println("A small loudspeaker appears in the air.");
        System.out.println("You hear the sound of a deep gong and a pleasant voice says:");
        System.out.println("Sorry, the correct answer is ");
        System.out.println(question.answer());
        System.out.println(".");
        loss = Math.min(this.tokens, adventurer.tokens());
        System.out.print("You see ");
        System.out.println(loss);
        System.out.print(" valuable tokens fly out of your pocket and fall to the floor.");
        System.out.println("A small vacuum cleaner appears, sweeps up your scattered tokens and disappears.");
        adventurer.loseTokens(loss);
        adventurer.reportTokens();
    }

import java.util.*;
public class Question {
    /*
       An instance of this class represents an arithmetic problem in the Arithmetic Adventure game.
       */

    /* Constructor */
    public Question() {
        return null;
    }
}
/* Initialize me so that I have two operands. */
this.leftOperand = Question.generator.next(Question.maxOperand);
this.rightOperand = Question.generator.next(Question.maxOperand);
}

/* Instance Methods */
public boolean ask() {
/* Pose myself. Return true if the user’s response was correct and false otherwise. */
Integer answer;
answer = null;
while (answer == null) {
    this.display();
    answer = Keyboard.in.readInteger();
}
return answer.intValue() == this.answer();
}

public int answer() {
/* Answer my correct answer. */
return this.leftOperand + this.rightOperand;
}

/* Private Static Variables */
private static final int maxOperand = 9;
private static final RandomInt generator = new RandomInt(2);
/* Private Instance Variables */
private int leftOperand;
private int rightOperand;

/* Private Instance Methods */
public void display() {
/* Display myself. */
System.out.print(this.leftOperand);
System.out.print(" + ");
System.out.print(this.rightOperand);
System.out.print(" = ");
}
}
import java.util.*;
import java.lang.*;
public class RandomInt {
  /*
   An instance of this class represents a generator that can generate a series of
   random positive ints.
   */
  /* Constructor */
  public RandomInt(int seed) {
    /* Initialize me so that I use the given seed. */
    this.generator = new Random(seed);
  }
  /* Instance Methods */
  public int next(int max) {
    /* Answer a Random int between 1 and the given max. */
    return Math.round(max * this.generator.nextFloat() - 0.5f) + 1;
  }
  /* Private Instance Variables */
  private Random generator;
}
import java.io.*;
import java.util.*;
public class TextMenu {
  /*
   An instance of this class displays a list of strings for the user
   and allows the user to pick one.
   */
  /* Constructor */
  public TextMenu() {
    /* Initialize me with no entries. */
    this.entries = new Vector();
  }
  /* Instance Methods */
  public void add(String entry) {
    /* Add the given String to me as my next choice. */
    this.entries.addElement(entry);
  }
  public String launch() {
    /* Display myself and answer the String entry selected
       by the user. */
    String       action;
int index;

index = this.getUserSelection();
action = (String) this.entries.elementAt(index - 1);
return action;

*/

private Vector entries;

private void display() {
   /*
   Display myself on the screen.
   */
   String entry;
   int index;
   System.out.println();
   System.out.println("Please type a number and press the Enter key:");
   for (index = 0; index < this.entries.size(); index++) {
      entry = (String) this.entries.elementAt(index);
      System.out.print(index + 1);
      System.out.print(" ");
      System.out.println(entry);
   }
} /* Private Instance Methods */

private int getUserSelection() {
   /*
   Query the user for an action and answer the index of the choice. If the
   user does not answer with a valid action, query again.
   */
   Integer choice;
   int index;
   index = 0;
   while (((index < 1) || (index > this.entries.size()))) {
      this.display();
      choice = Keyboard.in.readInteger();
      if (choice == null)
         index = 0;
      else
         index = choice.intValue();
   }
   return index;
} /* Private Instance Variables */

public class Door {
   /*
   An instance of this class represents a door in the Adventure game. A Door has a
   color and it connects two rooms.
   */
   /* Constructor */
   public Door(String color, Room aRoom, Room bRoom) {
      /*
      Initialize me so that I have the given color and
      connect the given two Rooms.
      */
this.color = color;
this.room1 = aRoom;
this.room2 = bRoom;
}

/* Instance Methods */

public void display() {
  /*
    Output a description of myself.
   */
  System.out.print("There is a ");
  System.out.print(this.color);
  System.out.println(" door in one wall.");
}

public String color() {
  /*
   Answer a String representing my color.
  */
  return color;
}

public Room adjoiningRoom(Room aRoom) {
  /*
    Answer the room that I connect the given Room to, or
    null if I don't connect it to any Room.
   */
  if (this.room1 == aRoom)
    return this.room2;
  else if (this.room2 == aRoom)
    return this.room1;
  else
    return null;
}

/* Private Instance Variables */

private String color;
private Room room1;
private Room room2;