Structural Programming and Data Structures

Winter 2000

CMPUT 102: Sharing Resources

Dr. Osmar R. Zaïane

University of Alberta

Course Content

Objectives of Lecture 13

Common Resources – Static Variables and Methods

- Understand the use of static variables to share common information between instances of the same class.
- Study the use of static methods to perform computations that independent of any object.
- Re-write the Adventure program using some useful static variables .

Outline of Lecture 13

- Static variables
- Static methods
- Adventure Version 4

Resources Common to a Class

- Sometimes common sharable resources are needed by each object in a class.
  - Each Circle object may need the value of π so it can compute and return its area.
  - Each TreasureChest object may need a common maximum number of tokens so it can generate a random number ≤ this number when it is constructed (initialized).
  - Each TreasureChest object may need a random number generator object to generate its initial number of tokens.

Static Variables

- It is wasteful to require every object in a class to use an instance variable to access these common resources.
- Instead, we can define a **static variable** in the class that can be bound to this common resource:

```
public class Circle …
   private static final float pi = 3.14159f;

public class TreasureChest …
   private static final int maxTokens = 10;
   private static RandomInt generator;
               generator = new RandomInt(1);
```

Outline of Lecture 13

- Static variables
- Static methods
- Adventure Version 4

Object Independent Actions

- Some actions are not performed by a specific instance of a class.
  - Start an application program.
  - Perform an operation on some values.
- In fact, no instance may even need to exist for these actions to be performed.

Static Methods

- A **static method** or **class method** is code that can be executed without sending a message to any object:

```
public class OurProgram …
    public static void main(String args[])
public class String …
    public static String valueOf(int i)
```

- The syntax of static method calls looks like a message is being sent to a class:

String.valueOf(3) ➔ “3”

- However, no message is actually involved.

Outline of Lecture 13

- Static variables
- Static methods
- Adventure Version 4

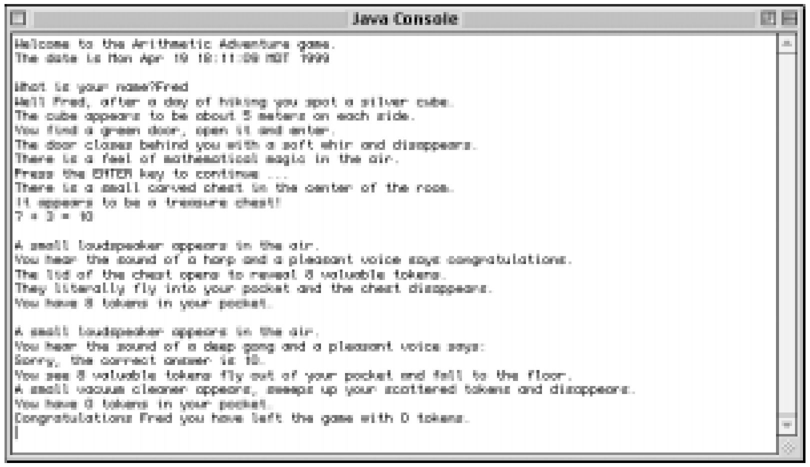
Adventure Version 4

- We are going to add some functionality to the Arithmetic Adventure game .
- We will put treasure chests in rooms.
- When the adventurer tries to open a chest we will generate an arithmetic question.
- 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.

Adventure - Code Change Summary

- 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.

Running Adventure 4



```
Welcome to the Arithmetic Adventure game.
The date is Mon Apr 19 10:11:09 MDT 1999

What is your name?Fred
Well Fred, after a day of hiking you spot a silver cube.
The cube appears to be about 5 meters on each side.
You find a green door, open it and enter.
The door closes behind you with a soft whir and disappears.
There is a feel of mathematical magic in the air.
Press the ENTER key to continue ...
There is a small carved chest in the center of the room.
It appears to be a treasure chest!
7 + 3 = 10

A small loudspeaker appears in the air.
You hear the sound of a harp and a pleasant voice says congratulations.
The lid of the chest opens to reveal 8 valuable tokens.
They literally fly into your pocket and the chest disappears.
You have 8 tokens in your pocket.

A small loudspeaker appears in the air.
You hear the sound of a deep gong and a pleasant voice says:
Sorry, the correct answer is 10.
You see 8 valuable tokens fly out of your pocket and fall to the floor.
A small vacuum cleaner appears, sweeps up your scattered tokens and disappears.
You have 0 tokens in your pocket.
Congratulations Fred you have left the game with 0 tokens.
```

Program - Adventure 4.1

NO CHANGES

```
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.
   */
   }
```

NO CHANGES

Program - Adventure 4.2

```
/* Main program */

   public static void main(String args[]) {
        Adventure       game;

        game = new Adventure();
        game.play();
   }
```

Program - Adventure 4.3

NO CHANGES

```
/* Private Instance Methods */

   private void play() {
   /*
        Plays the Adventure game.
   */

        Adventurer      adventurer;

        adventurer = this.greeting();
        this.enterRoom(adventurer);
        this.farewell(adventurer);
   }
```

Program - Adventure 4.4

NO CHANGES

```
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();
```

Program - Adventure 4.5

NO CHANGES

```
        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);
    }
```

~~Program - Adventure 3.6~~

~~OLD~~

~~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.readInteger();~~
~~adventurer.gainTokens(myTokens.intValue());~~
~~}~~

Program - Adventure 4.6

NEW

```
private void enterRoom(Adventurer adventurer) {
    /*
        The given adventurer has entered the
        first room.
    */
        Chest           chest;

        chest = new Chest();
        chest.display();
        chest.open(adventurer);
    }
```

Program - Adventure 4.7

NO CHANGES

```
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.");
}
```

Class - Chest 4.1

```
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);
   }
```

Class - Chest 4.2

```
/* 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!");
   }
```

Class - Chest 4.3

```
   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);
   }
```

Class - Chest 4.4

```
/* 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 */
```

Class - Chest 4.5

```
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();
  }
```

Class - Chest 4.6

```
    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.print(question.answer());
          System.out.println(".");
```

Class - Chest 4.7

```
      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();
  }
}
```

Class - Question 4.1

```
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);
    }
```

Class - Question 4.2

```
    /* Instance Methods */
    public void ask() {
    /*
          Pose myself. Eventually I would like to indicate
          whether the user's response was correct or not.
    */
          Integer answer;

          System.out.print(this.leftOperand);
          System.out.print(" + ");
          System.out.print(this.rightOperand);
          System.out.print(" = ");
          answer = Keyboard.in.readInteger();
    }
```

Class - Question 4.3

```
    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;
```

Class - RandomInt 4.1

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

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

Class - RandomInt 4.2

```
/* 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;
```

No Changes to End of Lecture

- The rest of the Adventure program is included for completeness.
- There are no changes from the last version in the rest of these slides.

Class - Adventurer 4.1

NO CHANGES

```
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;
   }
```

Class - Adventurer 4.2

NO CHANGES

```
/* 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;
   }
```

Class - Adventurer 4.3

NO CHANGES

```
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;
}
```

Class - Adventurer 4.4

NO CHANGES

```
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 */


public void reportTokens() {
private String name;
private int tokens;
```