한양대학교

OBJECT-ORIENTED SYSTEMS DESIGN (Lab6-2)

Heejin Park

Hanyang University

6-6 (Display 6.7)

Create classes *UtilityClass* and *VariableParameterDemo* defined as follows.

[*UtilityClass*]

1. Create a static method **public static int max(int... arg)**:

   If there is no argument,

   print out "Fatal Error: maximum of zero values." and exit.

   Create a variable **int** *largest* and store **arg[0]** in it.

   Store the largest among **arg[]** into *largest* by using a **for** statement.

   Return *largest.*

6-6

[*VariableParameterDemo*]

Write a class *VariableParameterDemo* that prints the input and output below using the class *UtilityClass*.

<input and output>

```
Enter scores for Tom, Dick, and Harriet:
55 100 99
Highest score = 100
```

6-7 (Display 6.8)

Create classes *Utility2* and *StringProcessingDemo* defined as follows.

[*Utility2*]

1. Create a static method **public static String censor(String sentence, String... unwanted):**

   It returns its first parameter **sentence** with all occurrences of the strings in **unwanted** being removed.

   (Use a **for** statement and the method **deleteOne.**)

6-7

2. Create a static method **private static String deleteOne(String sentence, String oneUnwanted):**

Create two variables **String** *ending* and **int** *position* with initializing *position* to **sentence.indexOf(oneUnwanted).**

While a word is found in **sentence**, store **sentence.substring(position + oneUnwanted.length())** into *ending*, **sentence.substring(0, position) + ending** into *sentence*, and **sentence.indexOf(oneUnwanted)** into *position*.

Return **sentence.**

6-7

[*StringProcessingDemo*]

Write a class *StringProcessingDemo* that prints the input and output below using the class *Utility2*.

<input and output>

```
What did you eat for dinner?
I ate salt cod, broccoli, french fries, salt peanuts, and apples.
You would be healthier if you could answer:
I ate  cod, broccoli,  peanuts, and apples.
```

← This output comes by deleting ("candy", "french fries", "salt", "beer") using the method **censor**.

6-8 (Display 6.9)

Create a class *ToyExample*.

Create a class *Date* by copying the class *Date* from chapter 4.

[*ToyExample*]

1. Create a instance variable **private Date[]** *a.*

2. Create a constructor **public ToyExmaple(int arraySize)** :

   Allocate to *a* an array **Date** of size **arraySize**.

   Allocate to *a[i]* a **new Date()** by using a **for** statement.

6-8

3. Create a constructor **public ToyExmaple(ToyExample object)** :

   Deep copy **object.a[]** to **a[]**.

4. Create a accessor method **public Date[] getDateArray():**

   Create a variable **Date[]** *temp*.

   Create a deep copy of **a[]** and make *temp* point to it.

   Return *temp*.

6-9 (Display 6.11, 6.12)

Create classes *SelectionSort* and *SelectionSortDemo* defined as follows.

[*SelectionSort*]

1. Create a static method **public static void sort(double[] a, int numberUsed):**

    Create two variables **int** *index* and *indexOfNextSmallest.*

    Sort array **a** by

    storing **indexOfSmallest(index, a, numberUsed)** into *indexOfNextSmallest*

    and invoking **interchange(index, indexOfNextSmallest, a)**.

6-9

2. Create a static method **private static int indexOfSmallest(int startIndex, double[ ] a, int numberUsed)**:

   Create three variables **double** *min* initialized to a[startIndex], **int** *indexOfMin* initialized to startIndex, and **int** *index.*

   Compare *min* with all the values of **a** and store the smallest value into *min* and its index into *indexOfMin* by using a **for** statement.

   Return *indexOfMin.*

3. Create a static method **private static void interchange(int i, int j, double[ ] a)**:

   Swap the values of **a[i]** and **a[j].**

6-9

[*SelectionSortDemo*]

Write a class *SelectionSortDemo* that prints the output below using the class *SelectionSort*.

<output>

```
Array contents before sorting:
7.7 5.5 11.0 3.0 16.0 4.4 20.0 14.0 13.0 42.0
Sorted array values:
3.0 4.4 5.5 7.7 11.0 13.0 14.0 16.0 20.0 42.0
```

6-10 (Display 6.13)

Create a class *EnumDemo* that prints the input and output below.

The detailed description of the class is given on the next page.

<output>

```
Work starts on MONDAY
Work ends on FRIDAY
```

6-10 (Display 6.13)

[*EnumDemo*]

1. List the day of five-days work week using the **enumerated type**. A value of an **enumerated type** is spelled with all uppercase letters.

   enum *WorkDay* {MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY};

2. Write a method **main**:

   1. Declare the variables **WorkDay type** *startDay* and *endDay* and store "MONDAY" and "FRIDAY" of **WorkDay type** in them.
   2. Print the output in the previous page.

6-11 (Display 6.15)

Create a class *EnumValuesDemo* that prints the input and output below.

The detailed description of the class is given on the next page.

<input and output>

```
Enter hours worked for MONDAY
8
Enter hours worked for TUESDAY
8
Enter hours worked for WEDNESDAY
8
Enter hours worked for THURSDAY
8
Enter hours worked for FRIDAY
7.5
Total hours worked = 39.5
```

6-11

[*EnumValuesDemo*]

1. List the five working days using the **enumerated type**. A value of an **enumerated type** is spelled with all uppercase letters.

    enum *WorkDay* {MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY};

2. Write a method **main**:
    1. Create an array **WorkDay type** *day* and store all values of *workday*.
    2. Create the variables **double** *hours* and *sum* and store 0 in them.
    3. Print the output in the previous page. (Use a **for** statements.)

6-12 (Display 6.16)

Create a class *EnumSwitchDemo* that prints the input and output below.

The detailed description of the class is given on the next page.

<input and output>

```
What is your favorite flavor?
Vanilla
Classic
```

<input and output>

```
What is your favorite flavor?
CHOCOLATE
Rich
```

<input and output>

```
What is your favorite flavor?
STRAWBERRY
I bet you said STRAWBERRY.
```

<input and output>

```
What is your favorite flavor?
PISTACHIO
```

← This input causes the program to end and issue an error message.

6-12

[*EnumSwitchDemo*]

1. List the kind of flavors using the **enumerated type**. A value of an **enumerated type** is spelled with all uppercase letters.

    enum *Flavor* {VANILLA, CHOCOLATE, STRAWBERRY};

2. Create a method **main**:
    1. Declare a variable **Flavor** *favorite* initialized to **null**.
    2. Create variables **String** *answer* and store the input.
    3. Print the output in the previous page. (Use a **switch** statements.)