Object Oriented Programming

Introduction to Java

Ch. 3 ~7

Dept. of Software, Gachon University
Ahyoung Choi, Spring

Homework2

- Due date: 10 days
- Assignment (total 160 pt, 20 pt each)
  - Chapter 3 Programming projects 3, 4
  - Chapter 4 Practice Programs 6, 11
  - Chapter 5 Programming projects 4, 8
  - Chapter 6 exercise 6&7
  - Chapter 7 exercise 20
- Submission form: **소스파일 제출(.java)**
- Submission site: 사이버 캠퍼스

HW 2-1

<Result>

```
Please enter a line to be checked for profanity
I am a doG.
Your input line
        does not contain cat
        contains dog
        does not contain llama
This line would be considered profane.
```

- Chapter 3 Programming projects 3
  - Suppose that we are working for an online service that provides a bulletin board for its users. We would like to give our users the option of filtering out profanity(욕설).
  - Suppose that we consider the words *cat*, *dog*, and *llama* to be profane.
  - Write a program that **reads a string from the keyboard** and **tests whether the string contains one of our profane words**. Your program should find words like *cAt* that differ only in case.

HW 2-2

<Result>

```
Please enter a date to be checked using mm/dd/yyyy/ format
12/80/2019/
date is 80 month 12 day 2019 year
It is not a valid date.
The reason it is invalid: The month value is not from 1 to 12.
```

- Chapter 3 Programming projects 4
  - Write a program that reads a string from the keyboard and tests **whether it contains a valid date(date only!)**. Display the date and a message that indicates whether it is valid. If it is not valid, also display a message explaining why it is not valid. The input date will have the format *mm*/*dd*/*yyyy*/.
  - A valid month value *mm* must be from 1 to 12 (January is 1).
  - The day value *dd* must be from 1 to a value that is appropriate for the given month. Assume that September, April, June, and November each have 30 days. February has 28 days except for leap years when it has 29. The remaining months all have 31 days each. A leap year is any year that is divisible by 4 but not divisible by 100 unless it is also divisible by 400.

HW 2-3

- Chapter 4 Practice programs 6
  - Write a program that read the temperature of a particular location at a particular time of day for each day of a week. Find an display the highest and the lowest values among the temperatures recorded. Also, find the average temperature of the location an display it.

Input	**Output**
21 30 25.8 27.3 24 29 22.7	Highest temperature is 30 Lowest temperature is 21 Average temperature is 26

HW 2-4

- Chapter 4 Programming projects 11
  - Your country is at ware and your enemies are using a secret code to communicate with one another. you have just learned that their encryption method is based upon the ASCII code. Individual characters in a string are encoded using this system.
  - Write a Java program the decrypts the intercepted message. You only know that the key used is a number between 1 and 100. And you assume that the original message consists entirely of ASCII codes that represent only printable characters.

For example, if the enemy uses key = 10 then the message “Hey” would be initially represented as follows. And “Hey” would be encrypted “Ro$”

Character	ASCII code	Encrypted value	Encrypted letter
H	72	82 (72+10)	R
e	101	111 (101+10)	O
y	121	36 (121+10)-127+32	$

HW 2-5

Listing 5.13 상관없이
입력 값 아래와 같이
초기화 하고 결과 구해도
ok

- Chapter 5 Programming projects 4
  - Write a program that uses the Purchase class in Listing 5.13 to set the following prices:
    - Oranges: 10 for $2.99
    - Eggs: 12 for $1.69
    - Apples: 3 for $1.00
    - Watermelons: $4.39 each
    - Bagels: 6 for $3.50
  - Then calculate the cost of each of the following five items and the total bill:
    - 2 dozen oranges
    - 3 dozen eggs
    - 20 apples
    - 2 watermelons
    - 1 dozen bagels

Listing 5.13

```
import java.util.Scanner;
public class Purchase {
    private String name;
    private int groupCount; //Part of a price, like the 2 in 2 for $1.99.
    private double groupPrice; //Part of a price, like the $1.99
    private int numberBought; //Number of items bought.
    public void setName (String newName)    {         name = newName;    }
    public void setPrice (int count, double costForCount)    {
        if ((count <= 0) || (costForCount <= 0))        {
            System.out.println ("Error: Bad parameter in setPrice.");
            System.exit (0);
        }
        else         {
            groupCount = count;
            groupPrice = costForCount;
        }
    }
    public void setNumberBought (int number)    {
        if (number <= 0)        {
            System.out.println ("Error: Bad parameter in
setNumberBought.");
            System.exit (0);
        }
        else            numberBought = number;
    }
```

```
public void readInput ()    {
        Scanner keyboard = new Scanner (System.in);
        System.out.println ("Enter name of item you are purchasing:");
        name = keyboard.nextLine ();
        System.out.println ("Enter price of item as two numbers.");
        System.out.println ("For example, 3 for $2.99 is entered as");
        System.out.println ("3 2.99");
        System.out.println ("Enter price of item as two numbers, now:");
        groupCount = keyboard.nextInt ();
        groupPrice = keyboard.nextDouble ();
        while ((groupCount <= 0) || (groupPrice <= 0))        {
            System.out.println (
                    "Both numbers must be positive. Try again.");
            System.out.println ("Enter price of item as two numbers.");
            System.out.println ("For example, 3 for $2.99 is entered as");
            System.out.println ("3 2.99");
            System.out.println ("Enter price of item as two numbers, now:"
            groupCount = keyboard.nextInt ();
            groupPrice = keyboard.nextDouble ();
        }
        System.out.println ("Enter number of items purchased:");
        numberBought = keyboard.nextInt ();
        while (numberBought <= 0)        { //Try again:
            System.out.println ("Number must be positive. Try again.");
            System.out.println ("Enter number of items purchased:");
            numberBought = keyboard.nextInt ();
        }
    }
    public void writeOutput ()    {
        System.out.println (numberBought + " " + name);
        System.out.println ("at " + groupCount +
                " for $" + groupPrice);
    }
    public String getName ()    {        return name;    }
    public double getTotalCost ()    {        return (groupPrice /
groupCount) * numberBought;    }
    public double getUnitCost ()    {        return groupPrice /
groupCount;    }
    public int getNumberBought ()    {        return numberBought;    }
}
```

HW 2-6

- 예시와 달라도 attribute 정의와 method 정의가 맞으면 ok
- 현장예약 및 전화예약 변수 및 함수 작성
- 한 개 이상의 티켓 구매 가능 여부 채크
- 판매 티켓 수 및 남은 티켓수 채크 함수 구현
- 콘서트의 총 판매량 채크 함수 구현
- Test case 작성

- Chapter 5 Programming projects 8
  - Consider a **class ConcertPromoter** that records the tickets sold for a performance. Before the day of the concert, tickets are sold only over the phone. Sales on the day of the performance are made only in person at the concert venue.
  - The class has the following attributes:
    - The name of the band
    - The capacity of the venue
    - The number of tickets sold
    - The price of a ticket sold by phone
    - The price of a ticket sold at the concert venue
    - The total sales amount
  - It has methods to
    - Record the sale of one or more tickets
    - Change from phone sales to sales at the concert venue
    - Return the number of tickets sold
    - Return the number of tickets remaining
    - Return the total sales for the concert

HW 2-7

- 메소드 누락시 감점
- RoomOccupancy 클래스 인스턴스 선언 등 활용 여부 확인
- Test 클래스 작성

- Chapter 6- exercise 6&7
- Create a class RoomOccupancy that can be used to record the number of people in the rooms of a building. The class has the attributes
  - numberInRoom—the number of people in a room
  - totalNumber—the total number of people in all rooms as a static variable
- The class has the following methods:
  - addOneToRoom—adds a person to the room and increases the value of totalNumber
  - removeOneFromRoom—removes a person from the room, ensuring that numberInRoom does not go below zero, and decreases the value of totalNumber as needed
  - getNumber—returns the number of people in the room
  - getTotal—a static method that returns the total number of people

HW 2-7 cont’

- Chapter 6- exercise 6&7
- Test class should include
  - Make more than two room instances RoomA and RoomB
  - Test add one to room, remove one to room
  - Compute total occupied room number

```
Add one to room a and three to room b.
Room a holds 1
Room b holds 3
Total in all rooms is 4
Remove one from both rooms.
Room a holds 0
Room b holds 2
Total in all rooms is 2
Remove two from room a.
Room a holds 0
Room b holds 2
Total in all rooms is 2
Create room c and add three to it.
Room a holds 0
Room b holds 2
Room c holds 3
Total in all rooms is 5
```

HW 2-8

- 2차원 어레이 선언 누락 감점
- Static 함수 미선언 감점
- Test 케이스 작성

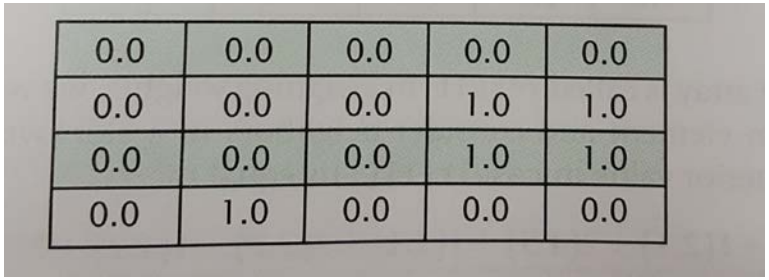
- Chapter 7 – exercise 20
- Write a static method findFigure(picture, threshold), where picture is a two-dimensional array of double values.
  - The method should return a new twodimensional array whose elements are either 0.0 or 1.0. Each 1.0 in this new array indicates that the corresponding value in picture exceeds threshold times(곱하기) the average of all values in picture. Other elements in the new array are 0.0.

HW 2-8 cont’

- 테이블 내 결과 오류 있습니다.
- 기본 로직만 맞으면 됩니다.
  - 평균*1.4 로 threshold 잡고 크면 1 적으면 0으로 작성

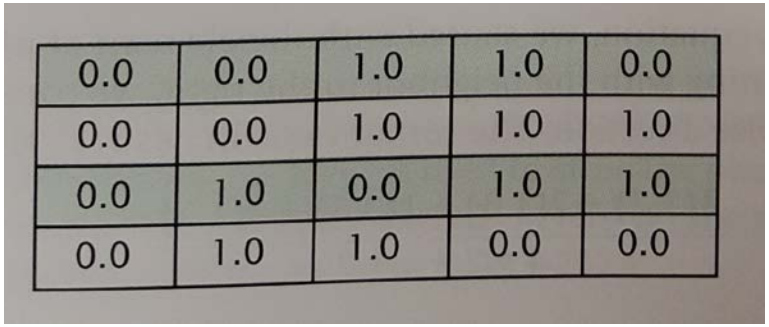
- For example, if the values in picture are the average value is 5.55.

- The resulting array for a threshold of 1.4 would be



0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	1.0
0.0	0.0	0.0	1.0	1.0
0.0	1.0	0.0	0.0	0.0

- Resulting array for a threshold of 0.6 would be



0.0	0.0	1.0	1.0	0.0
0.0	0.0	1.0	1.0	1.0
0.0	1.0	0.0	1.0	1.0
0.0	1.0	1.0	0.0	0.0