การทดลองที่ 6

Chapter 10 Object-Oriented Thinking

Program # 1 (Exercise 10.2 The BMI class) Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters.

// Construct a BMI with the specified name, age, weight, feet, and inches

public BMI(String name, int age, double weight, double feet, double inches)

BMI	Interpretation
BMI < 18.5	Underweight
18.5 ≤ BMI < 25.0	Normal weight
25.0 ≤ BMI < 30.0	Overweight
30.0 ≤ BMI	Obese

Write a program that prompts the user to enter name, age, weight, and height and displays the BMI together with the interpretation.

Note that one pound is 0.45359237 kilograms, and one inch is 0.0254 meters.

Sample run of program:

```
Enter name and age: Somsak 54 <enter>
Weight (input format: 1=kg 2=pound value): 1 65 <enter>
Height (input format: 1=meter 2=feet-inch): 2 5 10 <enter>
Your BMI: 20.56
Interpretation: normal weight
```

Program # 2 (Exercise 10.9 The Course class) Revise the Course class as follows:

- The array size is fixed in Listing 10.6. Improve it to automatically increase the array size by creating a new larger array and copying the contents of the current array to it.
- Implement the `dropStudent` method.
- Add a new method named `clear()` that removes all students from the course.

Write a test program that creates a course, adds three students, removes one, and displays the students in the course.

Program # 3 (Exercise 10.7 Game: ATM machine)

Use the Account class created in Programming Exercise 9.7 to simulate an ATM machine. Create ten accounts in an array with id 0, 1, ..., 9, and initial balance $100. The system prompts the user to enter an id. If the id is entered incorrectly, ask the user to enter a correct id. Once an id is accepted, the main menu is displayed as shown in the sample run. You can enter a choice 1 for viewing the current balance, 2 for withdrawing money, 3 for depositing money, and 4 for exiting the main menu. Once you exit, the system will prompt for an id again. Thus, once the system starts, it will not stop.

(See sample run of program on next page)

Sample run of program:

```
Enter account id: 4 <Enter>
Main menu
1: Check balance
2: Withdraw
3: Deposit
4: Exit
Enter your choice: 1 <Enter>
The balance is 100.00

Main menu
1: Check balance
2: Withdraw
3: Deposit
4: Exit
Enter your choice: 2 <Enter>
Enter amount to withdraw: 3 <Enter>
The balance is 97.00

Main menu
1: Check balance
2: Withdraw
3: Deposit
4: Exit
Enter your choice: 1 <Enter>
The balance is 97.00
```

```
Main menu
1: Check balance
2: Withdraw
3: Deposit
4: Exit
Enter your choice: 3 <Enter>
Enter amount to deposit: 10 <Enter>
The balance is 107.00

Main menu
1: Check balance
2: Withdraw
3: Deposit
4: Exit
Enter your choice: 1 <Enter>
The balance is 107.00

Main menu
1: Check balance
2: Withdraw
3: Deposit
4: Exit
Enter your choice: 4 <Enter>
Change account (y/n)?: n <enter>
End of Program.
```

Program # 4 (Exercise 10.10 The Queue class) Section 10.6 gives a class for Stack. Design a class named Queue for storing integers. Like a stack, a queue holds elements. In a stack, the elements are retrieved in a last-in first-out fashion. In a queue, the elements are retrieved in a first-in first-out fashion. The class contains:

- An int[] data field named `elements` that stores the int values in the queue.
- A data field named `size` that stores the number of elements in the queue.
- A constructor that creates a Queue object with default capacity 8.
- The method `enqueue(int v)` that adds v into the queue.
- The method `dequeue()` that removes and returns the element from the queue.
- The method `empty()` that returns true if the queue is empty.
- The method `getSize()` that returns the size of the queue.

Draw an UML diagram for the class. Implement the class with the initial array size set to 8. The array size will be doubled once the number of the elements exceeds the size. After an element is removed from the beginning of the array, you need to shift all elements in the array one position the left. Write a test program that adds 20 numbers from 1 to 20 into the queue and removes these numbers and displays them.
