

LAB 8

Class

Name : Lee Pei Hui

Matric Number : 22004781/1

Occurrence : 7

Lecturer : Dr. Soraya

1. Define a class name Number. The class is used to handle an integer array. The class Number will display all the item of the array, the even number, the prime number, the maximum number, the minimum number, the average, the square number. Create a Tester class to test the program.

Number a = new Number() // generate 10 random integers within 0 to 100.
Number b = new Number(5) // generate 5 random integers within 0 to 100.
Number c = new Number(4, 50) // generate 4 random integers within 0 to 50.

```
public class Number {  
  
    // Global Private Variable  
    private final int[] ARR;  
  
    // Constructors  
    public Number() {  
        ARR = new int[10];  
        for (int i = 0; i < 10; i++) { this.ARR[i] = (int) (Math.random() * 100); }  
    }  
  
    public Number(int n) {  
        ARR = new int[n];  
        for (int i = 0; i < n; i++) { this.ARR[i] = (int) (Math.random() * 100); }  
    }  
  
    public Number(int n, int ub) {  
        ARR = new int[n];  
        for (int i = 0; i < n; i++) { this.ARR[i] = (int) (Math.random() * ub); }  
    }  
  
    // Methods  
    public void displayAll() {  
        System.out.println("All Elements in Array:");  
        for (int elem: this.ARR) { System.out.printf("%d ", elem); }  
        System.out.println();  
    }  
  
    public void displayEven() {  
        System.out.println("All Even Elements in Array:");  
        for (int elem: this.ARR) { if (elem % 2 == 0) { System.out.printf("%d ", elem); } }  
        System.out.println();  
    }  
}
```

```
public void displayPrime() {
    boolean isPrime = false;
    System.out.println("All Prime Elements in Array:");
    for (int elem: this.ARR) {
        if (elem == 1 || elem == 0) { continue; }
        for (int i = 2; i < Math.sqrt(elem); i++) {
            if (elem % i == 0) { isPrime = false; break; }
            isPrime = true;
        }
        if (isPrime) { System.out.printf("%d ", elem); }
    }
    System.out.println();
}

public void displayMax() {
    int max = Integer.MIN_VALUE;
    for (int elem: this.ARR) { max = Math.max(max, elem); }
    System.out.printf("Largest Element in Array: %d\n", max);
}

public void displayMin() {
    int min = Integer.MAX_VALUE;
    for (int elem: this.ARR) { min = Math.min(min, elem); }
    System.out.printf("Smallest Element in Array: %d\n", min);
}

public void displayAvg() {
    int sum = 0;
    for (int elem: this.ARR) { sum += elem; }
    System.out.printf("Average of Elements in Array: %.2f\n", (sum + 0.0) / this.ARR.length);
}

public void displaySqNum() {
    System.out.println("All Elements Squared in Array:");
    for (int elem: this.ARR) { System.out.printf("%d ", elem * elem); }
    System.out.println();
}
}
```

2. Define a class name BankAccount. The class allows the customer to open the saving account with their name, IC or passport number and the deposit amount. The class consists of deposit method and withdraws method. Besides, the class can display the current balance. Create a Tester class to test the program.

```
public class BankAccount {  
  
    private final String NAME;  
    private final String IC;  
    private final String PASSPORT;  
    private double balance;  
  
    // Constructors  
    public BankAccount(String name, String ic, String passportNum, double deposit) {  
        this.NAME = name;  
        this.IC = ic;  
        this.PASSPORT = passportNum;  
        this.balance = deposit;  
        System.out.printf("Savings Account created for %s (IC Number [%s] and Passport Number [%s]).\n", this.NAME, this.IC, this.PASSPORT);  
        System.out.printf("Currently, there are RM%.2f in your account.\n", this.balance);  
    }  
  
    // Methods  
    public void deposit(double amount) {  
        this.balance += amount;  
    }  
  
    public void withdraw(double amount) {  
        if (amount < this.balance) { this.balance -= amount; }  
        else { System.out.println("The amount exceeds your current savings!"); }  
    }  
  
    public void displayBalance() {  
        System.out.printf("Amount in current savings: RM%.2f\n", this.balance);  
    }  
}
```

3. Define a class name WeightCalculator. The class has an input method that accepts the user's age and height. Besides, the class consists a method that calculating the recommend weight with this formula

$$\text{recommend weight} = (\text{height} - 100 + \text{age} / 10) * 0.9$$

The class will display the user's age, height and the recommend weight. Create a Tester class to test the program.

```
public class WeightCalculator {

    private final int AGE;
    private final double HEIGHT;

    // Constructor
    public WeightCalculator(int age, double height) {
        this.AGE = age;
        this.HEIGHT = height;
    }

    // Methods
    public double getRecommendedWeight() {
        return ((this.HEIGHT - 100) + ((double) this.AGE / 10)) * 0.9;
    }

    public void displayAge() {
        System.out.printf("Your age is %d.\n", this.AGE);
    }

    public void displayHeight() {
        System.out.printf("Your height is %.2fcm.\n", this.HEIGHT);
    }

    public void displayRecommendedWeight() {
        System.out.printf("Your recommended weight is %.2fkg.\n", getRecommendedWeight());
    }

}
```

4. Define a class Fraction. The class has an input method that accepts the numerator and the denominator from the user. Use the mutator method to set the numerator and denominator and the accessor method to get the value of numerator and denominator. This class also has a method to display the fraction reduced to lowest terms. (find the greatest common divisor for the numerator and denominator. Create a Tester class to test the program.

```
public class Main{
    public static void main(String[] args) throws Exception {
        Fraction fraction = new Fraction(100,400); //Create object from another class, Give parameters according to constructors
        System.out.println(fraction.getNumerator()+"-"+fraction.getDenominator());
        System.out.println(fraction);
        fraction.setDenominator(200);
        System.out.println(fraction);
        fraction.lowest_term();

    }
}

/*Define class Fraction*/
class Fraction{

    //Instance variable; access modifier of var: private [QTA]
    private int numerator,denominator;

    /*Input method that accepts value from user*/
    public Fraction (int numerator, int denominator) throws Exception {
        if(denominator == 0){
            throw new Exception("Denominator cannot be 0");
        }
        this.numerator = numerator;
        this.denominator = denominator;
    }

    /*Display Fraction reduced to lowest term*/
    private int gcd(int a, int b){
        if(b==0){return a;}
        else{return gcd(b,a%b);}
    }

    public void lowest_term(){
        int common_gcd = gcd(this.numerator, this.denominator);
        System.out.printf("%d/%d", (this.numerator/common_gcd), (this.denominator/common_gcd));
    }

    @Override
    public String toString(){ //For the get method
        return String.format("%d/%d", this.numerator, this.denominator);
    }
}
```

5. Define a class Game. The class has a constructor that accept player name. Besides, the class contains a method move that roll the dice. Create a Tester class to test the program with two players and the player that reach 100 or more win the game.

```
import java.util.Random;

public class Main{
    public static void main(String[] args){
        Game game = new Game("Ali","Bob");
        game.play(); //As play method accessed all other methods for the game, we only need to call it
    }
}

/*Define class Game*/
class Game{

    /*Instance variable*/
    //Access modifier: none
    String player1Name, player2Name;
    int player1Score, player2Score;

    //Access modifier: static
    static Random random = new Random();

    /*Constructor accept player name */
    public Game(String player1Name, String player2Name){ //without void & return type
        this.player1Name = player1Name;
        this.player2Name = player2Name;

        this.player1Score = player1Score;
        this.player2Score = player2Score;
    }

    public void play(){
        for(int i =0; i<202; i++){
            int roll = this.rollDice();
            if(i%2 != 0){//ODD: Player1 Turns
                this.player1Score += roll;
                System.out.printf("%s rolled %d\n", this.player1Name, roll);
            }else{ //EVEN: Player2 Turns
                this.player2Score += roll;
                System.out.printf("%s rolled %d\n", this.player2Name, roll);
            }
            System.out.printf("Current Score - %s:%d %s:%d\n\n", this.player1Name, this.player1Score, this.player2Name, this.player2Score);
            if(endGame()){
                break;
            }
        }
    }
}
```

```
/*Method move that roll dice*/
private int rollDice(){ //access modifier:
    return this.random.nextInt(6) + 1; //start from 1, count up 6 until 7 exclusive
}

private boolean endGame(){
    boolean player1Win = this.player1Score >= 100;
    boolean player2Win = this.player2Score >= 100;

    if (player1Win) {
        System.out.printf("%s has won!", this.player1Name);
    }
    if (player2Win) {
        System.out.printf("%s has won!", this.player2Name);
    }

    return player1Win || player2Win;
}

}

//Use this. to access methods (use methods in other method)
// Use this. to access variables (use variables in other method)
```


6. You operate several burger stalls distributed throughout town. Each burger stall has an ID and the number of burger sold for the day. The class consists of a constructor to initialize the instance variables and a sold method that increment the number of burger sold by parameter value. The class also contains a variable and method that tracks the total number of burgers sold in all stalls. Display the total burger sold by each stall and the total number of burgers sold in all stalls.

```
public class Main {
    public static void main(String[] args){
        BurgerStall burgerstall = new BurgerStall (new String[] {"a","b"}); //To initialize the object with parameter array (follow constructor)
        burgerstall.sold("a",20);
        burgerstall.sold("b",80);
        burgerstall.totalSold();
    }
}

/*BURGER STALL
- ID
- No. of burger sold for the day
*/
class BurgerStall{
    /* Instance Variable*/
    //Create an array as there are >1 stalls
    String[] stallID;
    int[] burgerSold;

    public BurgerStall (String[] stallID){
        this.stallID = stallID;
        this.burgerSold = new int[stallID.length]; //Array length same as stallID Array
    }

    /*Sold Method that increment number of burger sold by parameter value
    - Parameter pass in: StallID & num of burger sold
    - Matching the stall ID in array, if matched, will sum with the ori num stored
    */

    public void sold(String stallID, int burgersSold){
        for(int i = 0; i<this.stallID.length; i++){
            if(this.stallID[i].equals(stallID)){
                this.burgerSold[i] += burgersSold;
                break;
            }
        }
    }

    public void totalSold(){
        int totalSold = 0; //Must initialize this var in its method
        // Loop through array of burgerSold and add it to the totalSold
        for(int i = 0; i<this.stallID.length; i++) {
            totalSold += this.burgerSold[i];
            System.out.printf("Store %s sold %d burgers \n", this.stallID[i], this.burgerSold[i]);
        }
        System.out.println("Total Burgers sold: " + totalSold + " burgers");
    }
}
```

7. Define class money. The class is used to calculate the number of note and coins for a given amount of money. The available notes are RM100, RM50, RM10, RM5 and RM1. The available coins are 50cent, 20cent, 10cent and 5 cent. The class needs to

round up the amount according to the table below. The class also contains addition and subtraction method to add or subtract two objects.

Before Rounding	After Rounding
10.11, 10.12	10.10
10.13, 10.14, 10.16, 10.17	10.15
10.18, 10.19	10.20

```
public class Main{
    public static void main(String[] args)throws Exception{
        money money1 = new money(10,11);
        money money2 = new money(10,13);
        money money3 = new money(10,18);

        System.out.println(money1);
        System.out.println(money2);
        System.out.println(money3);

        money n1 = new money (10,80);
        money n2 = new money (14,11);
        n1.addition(n2);
        System.out.println(n1);
        n1.subtraction(n2);
        System.out.println(n1);
    }
}

class money{

    /* instance variable, access modifier = none */
    int cents;

    public money(int cents){
        this.cents = cents;
        this.round();
    }

    //Convert ringgit to cents
    public money(int ringgit, int cents){
        this.cents = ringgit * 100 + cents;
        this.round();
    }

    private void round(){
        int base = (this.cents/5)*5;
        int remainder = this.cents % 5;
        this.cents = base;
        if(remainder >= 3){
            this.cents += 5;
        }
    }
}
```

```
@Override
public String toString(){
    return String.format("RM%d.%02d", this.cents/100, this.cents%100);
}

public void subtraction(money other) {
    this.cents = this.cents - other.cents;
}

public void addition(money other) {
    this.cents = this.cents + other.cents;
}
}
```

Link:

<https://github.com/PeiHui369/Fundamentals-Of-Programming/tree/main/Lab%208/Self>