

Chapter 1 Java Review

1.1 Standard Input and Output

To output the standard output device, the monitor, there is the built-in class *System* that could be used. There is no need for an *import* statement. To read a datum from the standard input device, the keyboard, a common way is to use the *Scanner* class. An import statement for the *Scanner* class is needed.

Example Write a program that will prompt a message and read the user's full name from the keyboard, and then output a greeting message to the monitor.

```
import java.util.Scanner;
public class Greeting
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Type your name: ");
        String name = input.nextLine();
        System.out.println("Hello, " + name);
    }
}
```

Example Write a program that will prompt a message and read the radius of a circle from the keyboard, and then output its area to the monitor.

```
import java.util.Scanner;
public class CircleArea
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Type in the radius: ");
        double radius = input.nextDouble();
        double area = 3.14 * radius * radius;
        System.out.println("Area = " + area);
    }
}
```

Example Write a program that will prompt messages and read the length and width of a rectangle from the keyboard, and then output its area to the monitor.

```
import java.util.Scanner;
public class RectangleArea
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Type in the length: ");
        double length = input.nextDouble();
        System.out.print("Type in the width: ");
        double width = input.nextDouble();
    }
}
```

```
        double area = length * width;
        System.out.println("Area = " + area);
    }
}
```

Example Write a program that will prompt messages and read three words from the keyboard, and then output them in a reverse order to the monitor.

```
import java.util.Scanner;
public class ReverseWords
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Type the 1st word: ");
        String word1 = input.next();
        System.out.print("Type the 2nd word: ");
        String word2 = input.next();
        System.out.print("Type the 3rd word: ");
        String word3 = input.next();
        System.out.println(word3 + ", " + word2 + ", " + word1);
    }
}
```

With the redirection methods, the standard I/O could be used to handle the file. For example, assume that the source code of the program is *prog.java*, and after compilation, the byte code is created as *prog.class*; assume the input file as *in.txt* and the output file as *out.txt*. The command line is:

```
java prog < in.txt > out.txt
```

if the output file, *out.txt*, is existed before the execution, its content would be overwritten; otherwise, if it is not existed, it would be created automatically. In some cases, there is only input file, no output file, and the command line is:

```
java prog < in.txt
```

In some cases, there is only output file, no input file, and the command line is:

```
java prog > out.txt
```

Example Given an integer file, write a program that will read integers from the file, and then output them to the monitor or redirect the output to a file.

```
import java.util.Scanner;
public class ReadData
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        while(input.hasNextInt())
        {
            int data = input.nextInt();
            System.out.println(data);
        }
    }
}
```

Example Given an integer file, write a program that will read integers from the file, and then sum up them and calculate the average (display to two decimal places), and output to the monitor or redirect the output to a file.

```
import java.util.Scanner;
import java.text.DecimalFormat;
public class SumUpData
{
    public static void main(String[] args)
    {
        DecimalFormat f2 = new DecimalFormat("0.00");
        Scanner input = new Scanner(System.in);
        int data;
        int sum = 0;
        int size = 0;
        double average;
        while(input.hasNext())
        {
            data = input.nextInt();
            sum = sum + data;
            size ++;
        }
        average = (double)sum / size;
        System.out.println("Average = " + f2.format(average));
    }
}
```

Using an array for the above question as follows.

```
import java.util.Scanner;
public class ReadDataArray
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        final int MAX = 30;
        int[] data = new int[MAX];
        int size = 0;
        while(input.hasNext())
        {
            data[size] = input.nextInt(); // read data to an array
            size ++;
        }
        //System.out.println("Size = " + size);
        for(int i=0; i<size; i++)
            System.out.print(data[i] + " ");
        System.out.println("\n\nDone.\n\n");
    }
}
```

1.2 Classes

The user-defined money type is defined as a class shown below.

```
public class MoneyType
{
    private int dollars;
    private int cents;
    public void set(int d, int c)
    {    dollars = d;    cents = c; }
    public int getDollars()
    {    return dollars;    }
    public int getCents()
    {    return cents;    }
}
```

Example Based on the above definition of *MoneyType* class, write a program that will read the value of money in dollars and cents, declare an object, *money*, and then output the object values by invoking the methods of the class.

```
import java.util.Scanner;
public class MoneyTypeDemo
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Input dollars and cents - ");
        int d = input.nextInt();
        int c = input.nextInt();
        MoneyType money = new MoneyType();
        money.set(d, c);
        System.out.println("money = $" + money.getDollars() +
            "C" + money.getCents() );
    }
}
```