Jinan University

Java Programming Lab Report

Major: Computer Science & Technology
School: International School

Student name : _____
(p.s. your name on JNU academic system)

Student number : _____

Date of Submission (mm-dd-yyyy): _____

Instructor: Yuxia Sun

Table of Content

LAB 2 DATE: 3/14/2023

Student Name:_____ Student ID: _____

Problem 1. (3.5)

***3.5** (*Find future dates*) Write a program that prompts the user to enter an integer for today's day of the week (Sunday is 0, Monday is 1, . . . , and Saturday is 6). Also prompt the user to enter the number of days after today for a future day and display the future day of the week. Here is a sample run:

```
Enter today's day: 1 Enter
Enter the number of days elapsed since today: 3 Enter
Today is Monday and the future day is Thursday
```

**** Source Code / Solution :***

```
import java.util.Scanner;

public class FindFutureDates {
  final static String[] dateOfWeek = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};

  private static int calculator(int today, int gap) {
    return (today + gap) % 7;
  }

  private static String result(int today, int future) {
    return "Today is " +
        dateOfWeek[today] +
        " and the future day is " +
        dateOfWeek[future];
  }

  public static void main(String[] args) {
    int today, timeGap;
    Scanner input = new Scanner(System.in);

    System.out.print("Enter today's day: ");
```

```
        today = input.nextInt();
        System.out.print("Enter the number of days elapsed since today: ");
        timeGap = input.nextInt();

        System.out.println(result(today, calculator(today, timeGap)));
    }
}
```

**** Output:***

```
FindFutureDates ×
/Users/h3art/Library/Java/JavaVirtualMachines/op
Enter today's day: 1
Enter the number of days elapsed since today: 3
Today is Monday and the future day is Thursday

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: Using a complex simulation to calculate the date, the result was wrong.
Fix: The algorithm using modular arithmetic is found and the result is correct.

Problem 2. (3.31)

***3.31** (*Financials: currency exchange*) Write a program that prompts the user to enter the exchange rate from currency in U.S. dollars to Chinese RMB. Prompt the user to enter **0** to convert from U.S. dollars to Chinese RMB and **1** to convert from Chinese RMB to U.S. dollars. Prompt the user to enter the amount in U.S. dollars or Chinese RMB to convert it to Chinese RMB or U.S. dollars, respectively. Here are the sample runs:

```
Enter the exchange rate from dollars to RMB: 6.81 Enter
Enter 0 to convert dollars to RMB and 1 vice versa: 0 Enter
Enter the dollar amount: 100 Enter
$100.0 is 681.0 yuan
```

```
Enter the exchange rate from dollars to RMB: 6.81 Enter
Enter 0 to convert dollars to RMB and 1 vice versa: 1 Enter
Enter the RMB amount: 10000
10000.0 yuan is $1468.43
```

```
Enter the exchange rate from dollars to RMB: 6.81 ↵Enter
Enter 0 to convert dollars to RMB and 1 vice versa: 5 ↵Enter
CIncorrect input
```

**** Source Code / Solution :***

```
import java.util.Scanner;

public class CurrencyExchange {
   private static double CtoU(double rate, double amount) {
      return amount / rate;
   }

   private static double UtoC(double rate, double amount) {
      return amount * rate;
   }

   public static void main(String[] args) {
      double exchangeRate, amount;

      Scanner input = new Scanner(System.in);

      System.out.print("Enter the exchange rate from dollars to RMB: ");
      exchangeRate = input.nextDouble();

      System.out.print("Enter 0 to convert dollars to RMB and 1 vice versa: ");

      switch (input.nextInt()) {
         case 0:
            System.out.print("Enter the dollar amount: ");
            amount = input.nextDouble();
            System.out.printf("$%f is %f yuan\n", amount, UtoC(exchangeRate, amount));
            break;
         case 1:
            System.out.print("Enter the RMB amount: ");
            amount = input.nextDouble();
            System.out.printf("%f yuan is $%f\n", amount, CtoU(exchangeRate, amount));
            break;
         default:
```

```
            System.out.println("CIncorrect input");
        }
    }
}
```

**** Output:***

```
CurrencyExchange ×
/Users/h3art/Library/Java/JavaVirtualMachines/openjdk-
Enter the exchange rate from dollars to RMB: 6.81
Enter 0 to convert dollars to RMB and 1 vice versa: 0
Enter the dollar amount: 100
$100.000000 is 681.000000 yuan

进程已结束,退出代码0
CurrencyExchange ×
/Users/h3art/Library/Java/JavaVirtualMachines/openjdk-
Enter the exchange rate from dollars to RMB: 6.81
Enter 0 to convert dollars to RMB and 1 vice versa: 0
Enter the dollar amount: 100
$100.000000 is 681.000000 yuan

进程已结束,退出代码0
CurrencyExchange ×
/Users/h3art/Library/Java/JavaVirtualMachines/openjdk-
Enter the exchange rate from dollars to RMB: 6.81
Enter 0 to convert dollars to RMB and 1 vice versa: 5
CIncorrect input

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: I'm ignoring the non-0/1 case.
Fix: The calculation part was changed to switch judgment, and the incorrect input part was added.

Problem 3. (3.15)(Optional)

****3.15** (*Game: lottery*) Revise Listing 3.8, Lottery.java, to generate a lottery of a three-digit integer. The program prompts the user to enter a three-digit integer and determines whether the user wins according to the following rules:

1. If the user input matches the lottery number in the exact order, the award is $10,000.
2. If all digits in the user input match all digits in the lottery number, the award is $3,000.
3. If one digit in the user input matches a digit in the lottery number, the award is $1,000.

**** Source Code / Solution :***

```
import java.util.Scanner;

public class Lottery {
    private static void bubbleSort(int[] arr, int len) {
        for (int i = 0; i < len - 1; i++) {
            for (int j = 0; j < len - 1 - i; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }

    private static int randomInteger() {
        return (int) (Math.random() * 1000);
    }

    private static int[] partNumber(int num) {
        int[] result = new int[3];
        for (int i = 0; i < 3; i++) {
            result[i] = num % 10;
            num /= 10;
        }
        return result;
    }

    private static boolean singleMatch(int lottNum, int userNum) {
        int[] lott = partNumber(lottNum);
        int[] user = partNumber(userNum);
        for (int num1 : lott) {
            for (int num2 : user) {
                if (num2 == num1) {
                    return true;
                }
            }
        }
```

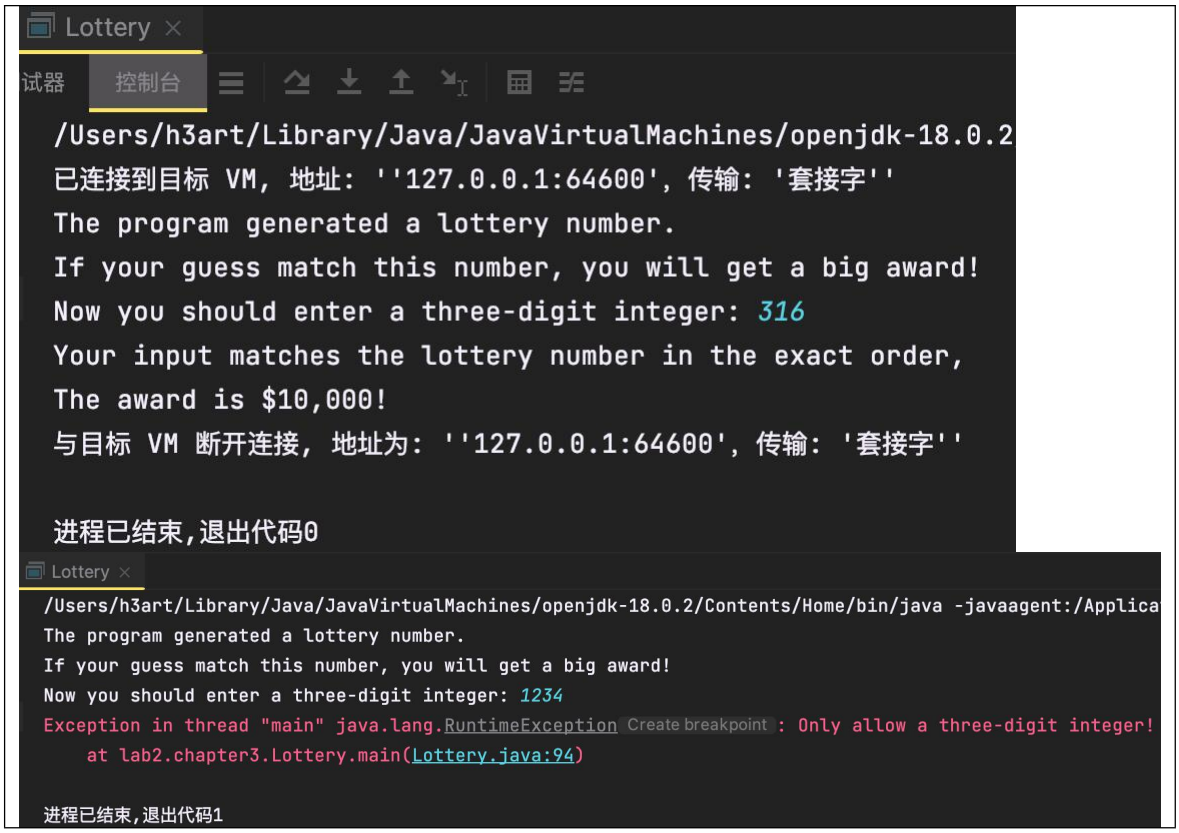
```
        return false;
    }

    private static boolean multiMatch(int lottNum, int userNum) {
        int[] lott = partNumber(lottNum);
        int[] user = partNumber(userNum);
        int len = lott.length;
        bubbleSort(lott, len);
        bubbleSort(user, len);
        for (int i = 0; i < len; i++) {
            if (lott[i] != user[i]) {
                return false;
            }
        }
        return true;
    }

    private static boolean completeMatch(int lottNum, int userNum) {
        return lottNum == userNum;
    }

    private static void awardJudge(int lottNum, int userNum) {
        if (singleMatch(lottNum, userNum)) {
            if (multiMatch(lottNum, userNum)) {
                if (completeMatch(lottNum, userNum)) {
                    System.out.println("Your input matches the lottery number in the exact
order,");
                    System.out.println("The award is $10,000!");
                } else {
                    System.out.println("All digits in your input match all digits in the
lottery number,");
                    System.out.println("The award is $3,000!");
                }
            } else {
                System.out.println("One digit in your input matches a digit in the lottery
number");
                System.out.println("the award is $1,000!");
            }
        } else {
            System.out.println("Your input didn't match any digit. What a shame!");
        }
    }

    public static void main(String[] args) {
        int lotteryNumber = randomInteger();
        int userNumber;

        Scanner input = new Scanner(System.in);

        System.out.println("The program generated a lottery number.");
        System.out.println("If your guess match this number, you will get a big award!");
        System.out.print("Now you should enter a three-digit integer: ");
```

```
        userNumber = input.nextInt();
        if (userNumber > 999 || userNumber < 100) {
            throw new RuntimeException("Only allow a three-digit integer!");
        }

        awardJudge(lotteryNumber, userNumber);
    }
}
```

**** Output:***

```
Lottery ×
/Users/h3art/Library/Java/JavaVirtualMachines/openjdk-18.0.2/
The program generated a lottery number.
If your guess match this number, you will get a big award!
Now you should enter a three-digit integer: 123
One digit in your input matches a digit in the lottery number
the award is $1,000!

进程已结束,退出代码0
```

```
Lottery ×
试器  控制台
/Users/h3art/Library/Java/JavaVirtualMachines/openjdk-18.0.2/Cont
已连接到目标 VM, 地址: ''127.0.0.1:64609', 传输: '套接字''
The program generated a lottery number.
If your guess match this number, you will get a big award!
Now you should enter a three-digit integer: 133
All digits in your input match all digits in the lottery number,
The award is $3,000!
与目标 VM 断开连接, 地址为: ''127.0.0.1:64609', 传输: '套接字''

进程已结束,退出代码0
```



**** Debugging/Testing:***

Bug1: No exception handling, resulting in numbers that do not conform to rules, can cause the program to run incorrectly.
Fix: Added data judgment, which will throw an exception to terminate the program if an error occurs.

Problem 4. (4.9)

***4.9** (*Find the Unicode of a character*) Write a program that receives a character and displays its Unicode. Here is a sample run:

```
Enter a character: E Enter
The Unicode for the character E is 69
```

**** Source Code / Solution :***

```
import java.util.Scanner;

public class FindUnicode {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
```

```
        String charStore;

        System.out.print("Enter a character: ");

        charStore = input.next();

        if (charStore.length() != 1) {
            throw new RuntimeException("Only allow 1 character!");
        }

        System.out.println("The Unicode for the character " + charStore.charAt(0)
                + " is " + charStore.codePointAt(0));

        input.close();
    }
}
```

**** Output:***

```
FindUnicode ×
/Users/h3art/Library/Java/JavaVirtualMachines
Enter a character: E
The Unicode for the character E is 69

进程已结束,退出代码0
FindUnicode ×
/Users/h3art/Library/Java/JavaVirtualMachines
Enter a character: -
The Unicode for the character - is 45

进程已结束,退出代码0
FindUnicode ×
/Users/h3art/Library/Java/JavaVirtualMachines
Enter a character: ¿
The Unicode for the character ¿ is 191

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: Using the %c placeholder of System.out.printf() directly causes the original input character, such as '¿', to fail to output properly will print 'Å'.
Fix: Change the character storage type to String, and use the string.codePointAt(int index) method to ensure correct

output.

Problem 5. (4.11)

***4.11** (*Decimal to hex*) Write a program that prompts the user to enter an integer between **0** and **15** and displays its corresponding hex number. For an incorrect input number, display invalid input. Here are some sample runs:

```
Enter a decimal value (0 to 15): 11 Enter
The hex value is B
```

```
Enter a decimal value (0 to 15): 5 Enter
The hex value is 5
```

```
Enter a decimal value (0 to 15): 31 Enter
31 is an invalid input
```

**** Source Code / Solution :***

```
import java.util.Scanner;

public class DeximalToHex {
    final static String hexValue = "0123456789ABCDEF";

    public static void main(String[] args) {
        int number;

        Scanner input = new Scanner(System.in);
        System.out.print("Enter a decimal value (0 to 15): ");

        number = input.nextInt();

        if (number >= 0 && number <= 15) {
            System.out.println("The hex value is " + hexValue.charAt(number));
            // System.out.printf("The hex value is %X\n", number);
        } else {
            System.out.println(number + " is an invalid input");
        }

        input.close();
    }
}
```

**** Output:***

```
DeximalToHex
/Users/h3art/Library/Java/JavaVirtualMachines
Enter a decimal value (0 to 15): 11
The hex value is B

进程已结束,退出代码0
DeximalToHex
/Users/h3art/Library/Java/JavaVirtualMachines
Enter a decimal value (0 to 15): 5
The hex value is 5

进程已结束,退出代码0
DeximalToHex
/Users/h3art/Library/Java/JavaVirtualMachines
Enter a decimal value (0 to 15): 31
31 is an invalid input

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: The single character in String object cannot be accessed directly using the [] operator.
Fix: The string.charAt(int index) method can be used to access a single character of the String.

Problem 6. (4.17)(Optional)

***4.17** (*Days of a month*) Write a program that prompts the user to enter the year and the first three letters of a month name (with the first letter in uppercase) and displays the number of days in the month. If the input for month is incorrect, display a message as presented in the following sample runs:

```
Enter a year: 2001 Enter
Enter a month: Jan Enter
Jan 2001 has 31 days
```

```
Enter a year: 2016 Enter
Enter a month: jan Enter
jan is not a correct month name
```

**** Source Code / Solution :***

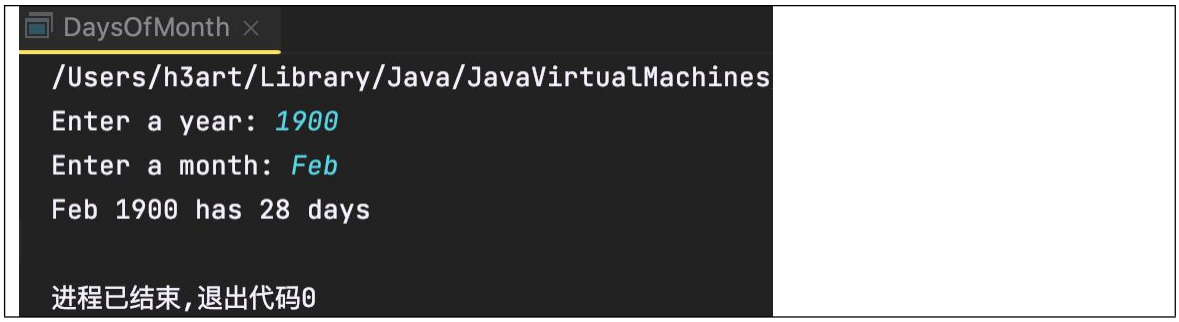
```
import java.util.HashMap;
```

```
import java.util.Objects;
import java.util.Scanner;

public class DaysOfMonth {

    final static HashMap<Integer, Integer> dayOfMonth = new HashMap<>(12);

    private static void initMap() {
        dayOfMonth.put(0, 31);
        dayOfMonth.put(1, 28);
        dayOfMonth.put(2, 31);
        dayOfMonth.put(3, 30);
        dayOfMonth.put(4, 31);
        dayOfMonth.put(5, 30);
        dayOfMonth.put(6, 31);
        dayOfMonth.put(7, 31);
        dayOfMonth.put(8, 30);
        dayOfMonth.put(9, 31);
        dayOfMonth.put(10, 30);
        dayOfMonth.put(11, 31);
    }

    final static String[] months = {"Jan", "Feb", "Mar", "Apr", "May", "Jun",
            "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};

    private static int match(String mon) {
        for (int i = 0; i < months.length; i++) {
            if (Objects.equals(months[i], mon)) {
                return i;
            }
        }
        return -1;
    }

    private static boolean isLeapYear(int year) {
        /*
        Either is the multiple of 400
        or is the multiple of 4 but not the multiple of 100
         */
        if (year % 400 == 0) {
            return true;
        } else return year % 100 != 0 && year % 4 == 0;
    }

    private static int calculator(int year, int mon) {
        if (mon == 1) {
            if (isLeapYear(year)) {
                return dayOfMonth.get(mon) + 1;
            }
        }
        return dayOfMonth.get(mon);
    }
```

```
    public static void main(String[] args) {
        int year;
        String month;
        initMap();

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a year: ");
        year = input.nextInt();
        System.out.print("Enter a month: ");
        month = input.next();

        int matchNum = match(month);
        if (matchNum == -1) {
            System.out.println(month + " is not a correct month name");
        } else {
            System.out.println(month + ' ' + year + " has " + calculator(year, matchNum) +
" days");
        }

        input.close();
    }
}
```

**** Output:***

```
DaysOfMonth ×
/Users/h3art/Library/Java/JavaVirtualMachines
Enter a year: 2001
Enter a month: Jan
Jan 2001 has 31 days

进程已结束,退出代码0
DaysOfMonth ×
/Users/h3art/Library/Java/JavaVirtualMachines
Enter a year: 2016
Enter a month: jan
jan is not a correct month name

进程已结束,退出代码0
```



**** Debugging/Testing:***

Bug1: Forget the uniform month store, because the index is computed from 0 in the String array store, resulting in access out of bounds.
Fix: Set all access/store month locations to start with the 0 index.