

【7】 次の各問いに答えなさい。

問1. プログラムの説明を読んで、プログラムの(1)~(2)を答えなさい。

<プログラムの説明>

処理内容

誕生日を入力し、12星座をディスプレイに表示する。

入力データ

| |
|-------------|
| 誕生日 ×××× |
|-------------|

(第1図)

実行結果

| |
|--|
| 誕生日の月日を入力してください 誕生日：1221 [星座は] いて座 |
| 誕生日：123 [星座は] みずがめ座 |
| 誕生日：801 [星座は] しし座 |
| 誕生日：0 |

(第2図)

処理条件

1. 配列 seizaBi に星座期間の最終日を、配列 seizaMei に星座名を記憶する。なお、各配列は添字で対応している。

配列

| | | | | | |
|----------|------------------|-------------------|---|---------------------|---------------------|
| seizaBi | (0) | (1) | ~ | (11) | (12) |
| | 119 | 218 | ~ | 1221 | 1231 |
| | [1月1日~ 1月19日] | [1月20日~ 2月18日] | ~ | [11月22日~ 12月21日] | [12月22日~ 12月31日] |
| seizaMei | (0) | (1) | ~ | (11) | (12) |
| | やぎ座 | みずがめ座 | ~ | いて座 | やぎ座 |

2. キーボードから第2図のように、誕生日の月日を入力する。なお、配列 seizaBi に記憶されている最終日および誕生日は、次の例のように構成されている。

例 307 → 3月7日

3. 誕生日をもとに配列 seizaBi を探索し、第2図のように表示する。

4. キーボードから誕生日に 0 が入力されたら処理を終了する。

<Javaプログラム>

//クラスSeiza

import java.util.Scanner;

```
public class Seiza {
    public static void main(String[] args) {
        int tanjoBi, i;
        int[] seizaBi = { 119, 218, ~ 1221, 1231 };
        String[] seizaMei = { "やぎ座", "みずがめ座", ~
            "いて座", "やぎ座" };
        Scanner sc = new Scanner(System.in);
        System.out.println("誕生日の月日を入力してください");
        System.out.print("誕生日：");
        (1) = sc.nextInt();
        while(tanjoBi > 0) {
            i = 0;
            while((2) < tanjoBi) {
                i = i + 1;
            }
            System.out.printf("[星座は] %-5s¥n", seizaMei[i]);
            System.out.println("");
            System.out.print("誕生日：");
            tanjoBi = sc.nextInt();
        }
        sc.close();
    }
}
```

問 2. プログラムの説明を読んで、プログラムの(3)~(5)を答えなさい。

<プログラムの説明>

処理内容

テーマパークの入場者数データを読み、アトラクション別とエリア別の入場者数をディスプレイに表示する。

入力データ

入場者数データ (ファイル名: nyujo.csv)

| 日 | アトラクションコード | エリアコード | 入場者数 |
|----|------------|--------|------|
| ×× | ×× | × | ×××× |

(第 1 図)

実行結果

| (アトラクション別入場者数) | | |
|----------------|---------|---------|
| (アトラクション名) | (入場者数計) | (人気度) |
| フォレスト A | 7,304 | ★★★ |
| ネイチャー 1 | 8,264 | ★★★★ |
| カヌーでゴー | 7,416 | ★★★ |
| ファイヤーフラッシュ | 3,962 | ★ |
| (エリア別入場者数) | | |
| (エリア名) | (入場者数計) | (1日の平均) |
| 森と炎 | 47,160 | 6,737 |
| 海と水 | 56,712 | 8,101 |
| 不思議 | 40,057 | 5,722 |
| キッズ | 15,531 | 2,218 |

(第 2 図)

処理条件

1. 第 1 図のアトラクションコードは 1~20, エリアコードは 1~4 であり, 1 週間分のデータが記録されている。

2. 配列 atorakuMei にアトラクション名を, 配列 eriaMei にエリア名を記憶する。なお, atorakuMei の添字はアトラクションコードと, eriaMei の添字はエリアコードと対応している。

配列

| | | | | | |
|------------|-----|---------|-----|--------|-----|
| atorakuMei | (0) | (1) | ~ | (20) | |
| | | フォレスト A | ~ | カヌーでゴー | |
| eriaMei | (0) | (1) | (2) | (3) | (4) |
| | | 森と炎 | 海と水 | 不思議 | キッズ |

3. テーマパークのエリアを管理する Eria クラスをインスタンス化し, 配列 eriaJoho に記憶する。なお, eriaJoho の添字は, エリアコードと対応している。

配列

| | | | | | |
|----------|-----|-----|-----|-----|-----|
| eriaJoho | (0) | (1) | (2) | (3) | (4) |
| | | | | | |

4. 第 1 図の入場者数データを読み, 入場者数を配列 atorakuKei に集計する。なお, atorakuKei の添字は, アトラクションコードと対応している。また, エリア別に入場者数を集計する。

配列

| | | | | |
|------------|-----|-----|---|------|
| atorakuKei | (0) | (1) | ~ | (20) |
| | | | ~ | |

5. データを読み終えたあと, アトラクション別入場者数を第 2 図のように表示する。なお, 人気度は, 2 千人ごとに「★」を表示する。

6. 最後に, エリアごとの 1 日の平均を次の計算式で求め, 第 2 図のように表示して処理を終了する。

$$1 \text{ 日の平均} = \text{入場者数計} \div 7$$

(注) Integer.parseInt(文字列型)
文字列型のデータを数値型(整数)に変換する。

<Javaプログラム>

```
//クラスEria
public class Eria {
    public String eriaMei;
    public int nyujoKei = 0;
    public Eria(String eriaMei) {
        this.eriaMei = eriaMei;
    }
    public void gokei(int nyujo) {
        nyujoKei = nyujoKei + nyujo;
    }
    public void hyoji() {
        int heikin = (3);
        System.out.printf("%-3s %7d %6d\n", eriaMei, nyujoKei, heikin);
    }
}

//クラスTemaPaku
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class TemaPaku {
    public static void main(String[] args) throws IOException {
        String[] atorakuMei = { "", "フォレスト A", "~", "カヌーでゴー" };
        String[] eriaMei = { "", "森と炎", "海と水", "不思議", "キッズ" };
        Eria[] eriaJoho = new Eria[5];
        int[] atorakuKei = new int[21];
        for(int h = 1; h <= 4; h++) {
            eriaJoho[h] = new Eria(eriaMei[h]);
        }
        for(int i = 1; i <= 20; i++) {
            atorakuKei[i] = 0;
        }
        BufferedReader fileIn = new BufferedReader(new FileReader("nyujo.csv"));
        String line;
        while((line = fileIn.readLine()) != null) {
            String[] str = line.split(",");
            int hi = Integer.parseInt(str[0]);
            int atorakuCode = Integer.parseInt(str[1]);
            int eriaCode = Integer.parseInt(str[2]);
            int nyujo = Integer.parseInt(str[3]);
            atorakuKei[atorakuCode] = (4);
            eriaJoho[eriaCode].gokei(nyujo);
        }
        fileIn.close();
        for(int i = 1; i <= 20; i++) {
            System.out.printf("%-14s %6d %6d", atorakuMei[i], atorakuKei[i]);
            int j = atorakuKei[i] / 2000;
            for(int k = 1; k <= j; k++) {
                System.out.print("★");
            }
            System.out.println("");
        }
        for(int h = 1; (5); h++) {
            eriaJoho[h].hyoji();
        }
    }
}
```