

10章 解答

1. 曜日と数を下の表のように対応付けます。

曜日	日	月	火	水	木	金	土
数	0	1	2	3	4	5	6

この表を使うと 1月1日から数えて (1月1日を1日目と数える) n 日目の曜日を計算することができます。その公式は、1月1日の曜日の数を m とすると次のようになります。

$$(m + n - 1) \% 7$$

例を示します。

1月1日が木曜日 ----- $m=4$

150日目は何曜日か --- $n=150$

公式から計算: $(4+150-1) \% 7 = 6 \rightarrow$ 表から「6」は「土曜日」である

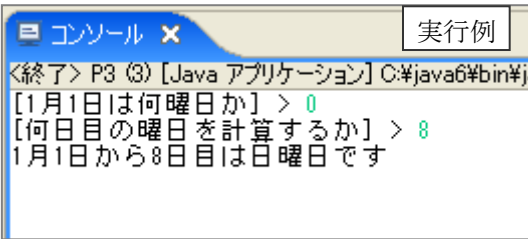
そこで、この表と公式を前提にして、次のようなプログラムを作成しなさい。

- ・1月1日の曜日に対応する数 m と、何日目かを表す n を入力する (`Input.getInt()` で)
- ・ n 日目の曜日に対応する数を公式から計算する
- ・表の対応関係を使って、`switch` 文により公式で求めた数を曜日に変換する
- ・曜日をコンソールに「1月1日から○日目は○曜日です」と表示する

【解答】 20 点

```
import lib.Input;
public class P11_3 {
    public static void main(String[] args) {
        int m = Input.getInt("1月1日は何曜日か");
        int n = Input.getInt("何日目の曜日を計算するか");
        String s;
        switch( (m+n-1)%7 ){
            case 0:
                s="日";
                break;
            case 1:
                s="月";
                break;
            case 2:
                s="火";
                break;
            case 3:
                s="水";
                break;
            case 4:
                s="木";
                break;
            case 5:
                s="金";
                break;
            default:
                s="土";
                break;
        }
        System.out.println("1月1日から" + n + "日目は" + s + "曜日です");
    }
}
```

Input クラスのメソッドは()内にガイドとして表示する文字列を書くことができます。下の実行例と見比べてください。



10章 解答

2. 次のプログラムを実行して 4 を入力したときの結果として正しいものを選択肢から選んで番号で答えなさい。

```
import lib.Input;
public class P11_5 {
    public static void main(String[] args) {
        int n=Input.getInt();// 4 を入力する
        switch(n%3.0){
            case 0:
            case 1:
                System.out.print("A");
                break;
            case 2:
            default:
                System.out.print("B");
        }
    }
}
```

《選択肢》

- ① A
- ② B
- ③ AB
- ④ コンパイルエラー
- ⑤ 実行時例外

【解答】

20 点

④

《解説》

$n\%3.0$ の計算では、結果は **double** になります。一般に式の中で異なる型が使われているとき、計算結果は最も自由度の高い型になります。 $n\%3.0$ は **int** と **double** の計算ですから結果は **double** になります。**double** は **switch** 文では使えません。

10章 解答

3. 右のプログラムについて、1～6の数値を入力したとき、どのように表示されるか、選択肢から選んで解答欄に答えなさい。

【解答】@3点 × 6 = 18点

入力	表示
1	1
2	2
3	13
4	1
5	4
6	1

《選択肢》

- | | |
|----------|----------|
| 1. A | 11. BCD |
| 2. B | 12. BCDE |
| 3. C | 13. CD |
| 4. D | 14. CDE |
| 5. E | 15. DE |
| 6. AB | |
| 7. ABC | |
| 8. ABCD | |
| 9. ABCDE | |
| 10. BC | |

```
int n=Input.getInt();
switch(n%5){
    case 1:
    case 4:
        System.out.print("A");
        break;
    default:
        System.out.print("B");
        break;
    case 3:
    case 6:
        System.out.print("C");
    case 0:
        System.out.print("D");
        break;
    case 5:
        System.out.print("E");
}
```

《解説》

switch 文では、処理の最後に **break** 文がないと次の **case** ラベル以下にある処理文を実行してしまいます。**case** 文はストッパーにならないことに注意してください。

また、**case** 文は好きな順序で並べることができます。**default** 文も最後ではなく、このように **switch** 文の中の好きな位置に書くことができます。

4. 次の **case** 文で正しいものに○、正しくないものに×を付けなさい。
ただし、`int m=1; char a='c';` とする。

@2点 × 11 = 22点

- [×] `case 3;`
- [×] `case m:`
- [×] `case m%3:`
- [×] `case 10.5:`
- [○] `case 'a':`
- [×] `case 10L:`
- [×] `case a+1:`
- [×] `case a>0:`
- [×] `case a%30:`
- [○] `case 10+1:`
- [×] `default 3:`

10章 解答

5. 次の switch 文の () 内にも書いてもよいものに○、書けないものに×を付けなさい。
ただし、int m=1; char a='c'; double x=2.0; byte t=0x0f;

```
switch( ){  
    case 1:  
    case 2:  
    default:  
}
```

@2 点 × 10 = 20 点

```
[×] 10F  
[○] m  
[○] (int)x  
[○] m%10  
[×] m-0.5  
[○] a  
[×] 10/x  
[○] 5  
[○] t  
[○] m+'¥u0048'
```