

(誤記の訂正をしています.)

(解答の注意)

- 1 各プログラムの先頭行に, 学籍番号と氏名をコメントで入力してください.
- 2 プログラムは, 問題毎に指定されたファイル名で保存してください.
- 3 完成したプログラムは, USBメモリに保存し, 問題用紙といっしょに提出してください.
- 4 試験時間は65分です.

**問題1 クラス図からソースプログラムの導出 (提出 MilkBar.java)**

クラス図 MilkBar からソースプログラムを導出しなさい.

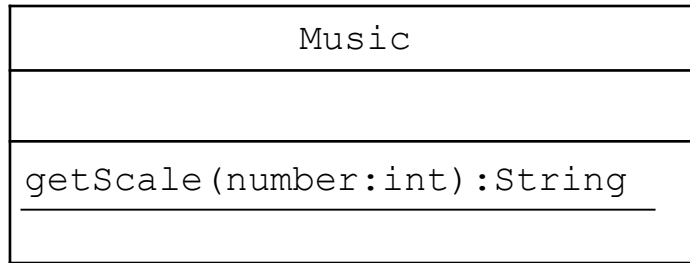
MilkBar
<u>welcome():void</u> <u>getName(number:int):String</u> <u>getPrice(name:String):int</u>

※milk bar : ミルクスタンド(牛乳やアイスクリームを売る店)

## 問題2 音階 (提出 Music.java)

このプログラムは, 音楽の音階(ドレミ..)を扱います. (1) ~ (3) の手順にしたがって, プログラムを作成しなさい.

(1) クラス図 Music からソースプログラムを導出しなさい.



(2) 動作確認用プログラム MusicMain.java を作成しなさい.

```
public class MusicMain{
    public static void main(String[] args){
        for(int i = 0; i < 8; i++){
            int n = i % 7;
            String scale = Music.getScale(n);
            System.out.println(i + "->" + scale);
        }
    }
}
```

「i % 7」の記述について  
iを7で割った余りを計算しています.

### 実行イメージ

```
> java MusicMain
0->ド
1->レ
2->ミ
3->ファ
4->ソ
5->ラ
6->シ
7->ド
```

(3) API仕様 Music を満足するように Music.java を実装しなさい.

API仕様 Music	
getScale	番号から音階の名称 (文字列) に変換し, 返却します.  引数 (number:int) は番号です.  番号と文字列の関係は, 以下の通りです. 0:ド, 1:レ, 2:ミ, 3:ファ, 4:ソ, 5:ラ, 6:シ

### 問題3 電話番号 (提出 Telephone.java)

このプログラムは、電話番号を扱います。(1)～(3)の手順にしたがって、プログラムを作成しなさい。

(1) クラス図 Telephone からソースプログラムを導出しなさい。

Telephone
<pre>getNumber(str1:String, str2:String, str3:String):String getSeparator(length:int, pattern:String):String</pre>

(2) 動作確認用プログラム TelephoneMain.java を作成しなさい。

```
public class TelephoneMain{  
    public static void main(String[] args){  
        String sep1 = Telephone.getSeparator(12, "=");  
        String sep2 = Telephone.getSeparator(3, "-");  
        String num1 = Telephone.getNumber("03", "5284", "5120");  
        String num2 = Telephone.getNumber("0476", "46", "4111");  
        System.out.println(sep1);  
        System.out.println(num1);  
        System.out.println(sep2);  
        System.out.println(num2);  
        System.out.println(sep1);  
    }  
}
```

#### 実行イメージ

```
> java TelephoneMain  
=====  
03-5284-5120  
---  
0476-46-4111  
=====
```

(3) API仕様 Telephone を満足するように Telephone.java を実装しなさい。

API仕様 Telephone	
getNumber	<p>電話番号(文字列)を作り、返却します。</p> <p>引数(str1:String)は、番号の最初の文字列です。 引数(str2:String)は、番号の中央の文字列です。 引数(str3:String)は、番号の最後の文字列です。</p> <p>電話番号を str1 "-" str2 "-" str3の順で連結し、返却します。</p>
getSeparator	<p>区切り線(文字列)を作り、返却します。</p> <p>引数(length:int)は、区切り線の長さ(繰り返しの数)です。 引数(pattern:String)は、区切り線に使う文字列です。</p> <p>引数(length:int)の数だけ、引数(pattern:String)の文字列を連結します。できあがった文字列を返却します。</p>

#### 問題4 図形 (提出 Shape.java)

このプログラムは, 図形を扱います. (1) ~ (4) の手順にしたがって, プログラムを作成しなさい.

(1) クラス図 Shape からソースプログラムを導出しなさい.

Shape
<u>getPart(s:String, i:int):String</u>
<u>makeLine(style:String, length:int):String</u>
<u>lineSeparator():String</u>
<u>reverse(s:String):String</u>
<u>mirror(s:String):String</u>
<u>makeTriangle(str:String, size:int):String</u>
<u>makeMountain(str:String, size:int):String</u>
<u>makeDiamond(str:String, size:int):String</u>

(2) プログラム ShapeTest1.java, ShapeTest2.java を作成し, 実行しなさい.

```
public class ShapeTest1{
    public static void main(String[] args){
        String s = "abcd";
        System.out.println(s.length());
        for(int i = 0; i < s.length(); i++) {
            String s1 = s.substring(i, i + 1);
            System.out.println(s1);
        }
    }
}
```

s.length() は,  
文字列sの文字数を返します

s.substring(i, i + 1)は,  
文字列sのi番目の1文字を返します。  
ただし, iは0から始まります。

実行イメージ

```
> java ShapeTest1
4
a
b
c
d
```

```
public class ShapeTest2{
    public static void main(String[] args){
        String s = "abcd";
        String s2 = "";
        for(int i = 0; i < s.length(); i++) {
            String s1 = s.substring(i);
            s2 = s2 + s1 + "¥n";
        }
        System.out.println(s2);
    }
}
```

s.substring(i)は,  
文字列sのi番目以降の文  
字列を返します。ただし, i  
は0から始まります。

s2 = s2 + s1 + "¥n"は,  
文字列s1と改行を, 文字列s2に追加します。

実行イメージ

```
> java ShapeTest2
abcd
bcd
cd
d
```

(3) 動作確認用プログラム ShapeMain.java を作成しなさい。

```
public class ShapeMain{
    public static void main(String[] args){
        // (1) getPart
        String s1 = "abcd";
        for(int i = 0; i < s1.length(); i++){
            String s = Shape.getPart(s1, i);
            System.out.println(s);
        }
        System.out.println();
        // (2-1) makeLine
        int size = 5;
        for(int i = 0; i < size; i++) {
            String s = Shape.makeLine("a", i + 1);
            System.out.println(i + ":" + s);
        }
        System.out.println();
        // (2-2) makeLine 2
        for(int i = 0; i < size; i++) {
            int j = size - i - 1;
            String s = Shape.makeLine(" ", j);
            s = s + Shape.makeLine("b", i + 1);
            System.out.println(i + ":" + s);
        }
        System.out.println();
        // (3) lineSeparator
        String s3 = "";
        for(int i = 0; i < size; i++) {
            String s = Shape.makeLine("c", i + 1);
            s3 = s3 + s + Shape.lineSeparator();
        }
        System.out.println(s3);
        // (4) reverse
        String s4 = Shape.reverse(s1);
        System.out.println("reverse:" + s4);
        System.out.println();
        // (5) mirror
        String s5 = Shape.mirror(s1);
        System.out.println("mirror:" + s5);
        System.out.println();
        // (6) triangle
        String s6 = Shape.makeTriangle("T", size);
        System.out.println(s6);
        // (7) mountain
        String s7 = Shape.makeMountain("M", size);
        System.out.println(s7);
        // (8) diamond
        int size2 = 8;
        String s8 = Shape.makeDiamond("D", size2);
        System.out.println(s8);
    }
}
```

実行イメージ

```
> java ShapeMain
a
b
c
d

0:a
1:aa
2:aaa
3:aaaa
4:aaaaa

0:   b
1:  bb
2: bbb
3: bbbb
4: bbbbb

c
cc
ccc
cccc
ccccc

reverse:dcba

mirror:abcdcba

  T
 TT
 TTT
 TTTT
 TTTTT

    MM
   MMM
  MMMMM
 MMMMMMM
MMMMMMMMMM

  DD
 DDDD
 DDDDDD
 DDDDDDDD
 DDDDDDDD
 DDDDDD
  DDDD
   DD
```

(4) API仕様 Shape を満足するように Shape.java を実装しなさい。

API仕様 Shape	
getPart	文字列sのi番目の文字(1文字)を取り出し、返却します。 ただし、iは0から始まります。 引数(s:String)は、文字列です。 引数(i:int)は、取り出す位置です。
makeLine	文字列styleをlengthの数だけ連結した文字列(line)を作成し、返却します。 引数(style:String)は、lineに使う文字列です。 引数(length:int)は、lineの長さです。
lineSeparator	改行コード(line separator)を返却します。 “\n”を返却します。
reverse	文字列sを逆順にした文字列を作成し、返却します。 引数(s:String)は、文字列です。
mirror	文字列sを逆順にした文字列を作成し、さらに、文字列sの後に結合し、返却します。 引数(s:String)は、文字列です。
makeTriangle	文字列で、三角形(直角三角形)を作成し、返却します。 引数(str:String)は、図形に使用する文字列です。 引数(size:int)は、図形の縦の大きさ(行数)です。  (実行イメージは、strが“T”, sizeが5の場合です。右下部分が直角に相当します。各行の最後に改行コードを使用しています。三角形の底辺の文字数はsizeと同じです。)
makeMountain	文字列で、山の形を作成し、返却します。 引数(str:String)は、図形に使用する文字列です。 引数(size:int)は、図形の縦の大きさ(行数)です。  (実行イメージは、strが“M”, sizeが5の場合です。各行の最後に改行コードを使用しています。左右対称になります。左右対称にするためにreverseメソッドを使用しても良い。)
makeDiamond	文字列で、ひし形(ダイヤ)を作成し、返却します。 引数(str:String)は、図形に使用する文字列です。 引数(size:int)は、図形の縦の大きさ(行数)です。ただし、偶数のみ扱います。  (実行イメージは、strが“D”, sizeが8の場合です。各行の最後に改行コードを使用しています。左右対称、上下対称になります。左右対称にするためにreverseメソッドを使用しても良い。)