

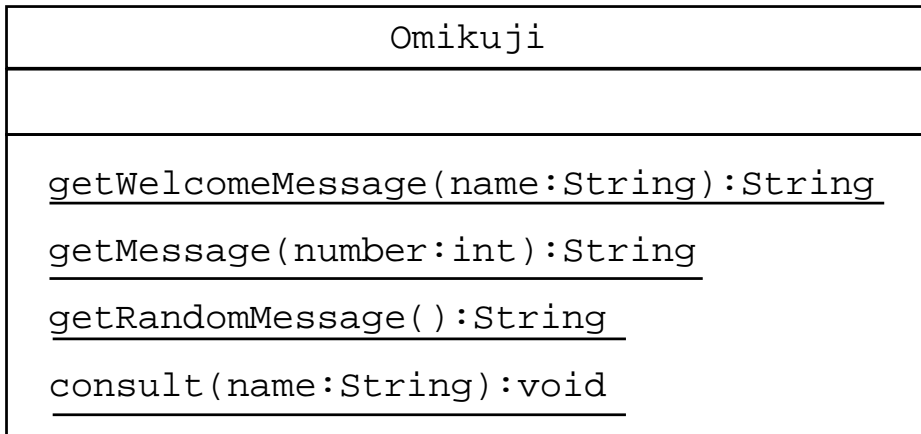
学籍番号:

氏名:

問題4 おみくじ

このプログラムは、おみくじを引いて、その結果を表示するものです。(1)~(3)に従って作成しなさい。

(1) クラス図 Omikuji 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 OmikujiTest.java と OmikujiMain.java を作成しなさい。

実行イメージ

```
> java OmikujiTest
Pro Taroさんの今日の運勢は
大吉
中吉です
```

実行イメージ

```
> java OmikujiMain
Pro Taroさんの今日の運勢は吉です
```

```
public class OmikujiTest{
    public static void main(String[] args){
        String welcome = Omikuji.getWelcomeMessage("Pro Taro");
        System.out.println(welcome);
        String fortune = Omikuji.getMessage(0);
        System.out.println(fortune);
        String message = Omikuji.getRandomMessage();
        System.out.println(message);
    }
}
```

```
public class OmikujiMain{
    public static void main(String[] args){
        Omikuji.consult("Pro Taro");
    }
}
```

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

API仕様 Omikuji	
getWelcome-Message	メッセージを返却します。 引数(name:String)は名前です。 名前と「さんの今日の運勢は」の文字列を連結し、返却します。
getMessage	おみくじの番号からおみくじ(文字列)に変換し、返却します。 おみくじの番号(number:int)とおみくじの関係は、以下の通りです。 0:大吉, 1:中吉, 2:小吉, 3:吉, 4:末吉, 5:凶, 6:大凶
getRandom-Message	おみくじの結果を返却します。 乱数によっておみくじの番号(0~6の整数)を生成します。 次におみくじの番号からおみくじの文字列(「大吉」等)に変換します。 おみくじの文字列と「です」を連結し、返却します。
consult	おみくじを引き、その結果を出力(表示)します。 引数(name:String)は名前です。 おみくじの結果を生成します。 名前と「さんの今日の運勢は」とおみくじの結果を出力(表示)します。

※ 乱数 RandomNumber クラス を利用するには以下の記述が必要です。
`import jp.tdu.util.RandomNumber;`

学籍番号:

氏名:

問題5 あっちむいて・ホイ

このプログラムは、あっちむいて・ホイのゲームです。(1)~(3)に従って作成しなさい。

(1) クラス図 Acchimuite からソースプログラムを導出しなさい。

Acchimuite
<pre>getMessage(number:int):String printIntroduction():void getPlayerNumber():int getComputerNumber():int getYouVsCom(player:int, computer:int):String getResult(player:int, computer:int):String hoi():void</pre>

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

実行イメージ

```
> java AcchimuiteTest
0:↑
1:→
2:↓
3:←
番号を入力してください: 0 Enter
player:0
computer:1
message:You:↑vs→:Com
```

実行イメージ

```
> java AcchimuiteMain
0:↑
1:→
2:↓
3:←
番号を入力してください: 2 Enter
You:↓vs↓:Com
You Win!!
```

```
public class AcchimuiteTest{
    public static void main(String[] args){
        Acchimuite.printIntroduction();
        int player = Acchimuite.getPlayerNumber();
        System.out.println("player:" + player);
        int computer = Acchimuite.getComputerNumber();
        System.out.println("computer:" + computer);
        String message = Acchimuite.getYouVsCom(player, computer);
        System.out.println("message:" + message);
    }
}
```

```
public class AcchimuiteMain{
    public static void main(String[] args){
        Acchimuite.hoi();
    }
}
```

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

API仕様 Acchimuite	
getMessage	番号を矢印(文字列)に変換し, 返却します. 引数(number:int)は番号です. 番号と矢印の関係は, 以下の通りです. 0:↑, 1:→, 2:↓, 3:←
print- Introduction	番号と矢印の対応を出力(表示)します. 0:↑ 1:→ 2:↓ 3:←
getPlayer- Number	(1)入力を促すメッセージ「番号を入力してください:」を出力(表示)します. (2)キーボードから番号を入力し, 返却します.
getComputer- Number	乱数を使用して, 番号を返却します. 番号は0~3の整数です.
getYouVsCom	プレイヤーとコンピュータの矢印を文字列として返却します. 引数(player:int)はプレイヤーの選んだ番号です. 引数(computer:int)はコンピュータの選んだ番号です. 「You:», playerの矢印, 「vs」, computerの矢印, 「:Com」の文字列を連結し, 返却します.
getResult	勝敗結果を文字列として, 返却します. 引数(player:int)はプレイヤーの選んだ番号です. 引数(computer:int)はコンピュータの選んだ番号です. player と computer が一致した場合は, 文字列「You Win!!」を返却します. それ以外は, 文字列「You Lose!!」を返却します.
hoi	ゲームをします. (他の振る舞いを呼び出して実現します.) (1)番号と矢印の対応を出力(表示)します. (2)プレイヤーの番号を入力します. (3)コンピュータの番号を決めます. (4)プレイヤーとコンピュータの選んだ矢印を出力(表示)します. (5)勝敗結果を出力(表示)します.

※ KeyboardReader クラス を利用するには以下の記述が必要です.
`import jp.tdu.util.KeyboardReader;`

※ RandomNumber クラス を利用するには以下の記述が必要です.
`import jp.tdu.util.RandomNumber;`