

特別プログラム クイズゲームを通してOOPを学ぶ

処理内容

問題を出題し、回答をキーボードから入力する。問題は全部で3問で、出題回数は5回とする。5回出題し終わったら、正解数を表示してプログラムを終了する。

※本来は、3問しかない問題を5回の中で出題するのは不自然だが、プログラムを簡略化するためそのような処理とした。

実行結果

```
***** 第 1 問 *****
10進数の15を2進数4桁にすると？
1101
不正解です
***** 第 2 問 *****
犬を英語で言うと？
dog
正解です！
***** 第 3 問 *****
犬を英語で言うと？
dog
正解です！
***** 第 4 問 *****
犬を英語で言うと？
dog
正解です！
***** 第 5 問 *****
1秒間に転送できるデータのビット数の単位は？
pps
不正解です
正解数は 3 問です
```

プログラム1 全てメインメソッドで処理 (OOPではない)

```
1 import java.util.Scanner;
2 public class NotObject2 {
3     public static void main(String[] args) {
4         int seikaisuu = 0;           //正解数
5         int syutudai;               //出題する問題の添字
6         String kaitou;              //プレイヤーが回答した答え
7         String[] mondai = new String[3];
8         String[] kotae = new String[3];
9         mondai[0] = "犬を英語で言うと?";
10        mondai[1] = "10進数の15を2進数4桁にすると?";
11        mondai[2] = "1秒間に転送できるデータのビット数の単位は?";
12        kotae[0] = "dog";
13        kotae[1] = "1111";
14        kotae[2] = "bps";
15        Scanner sc = new Scanner(System.in);
16        //5問出題する繰り返し処理
17        for(int kaisu = 1; kaisu < 6; kaisu++){
18            System.out.printf("***** 第 %d 問 *****\n", kaisu);
19            syutudai = (int) (Math.random() * 3);
20            System.out.println( _____ );
21            kaitou = sc.next();
22            if(kotae[syutudai].equals(kaitou)){
23                System.out.println("正解です!");
24                _____;
25            }else{
26                System.out.println("不正解です");
27            }
28        }
29        sc.close();
30        System.out.printf("正解数は %d 問です", seikaisuu);
31    }
32 }
```

プログラム2 メインメソッドとインスタンスメソッドで処理（OOPではない）

※全商検定2級のような形、サブルーチン化して保守性や可読性を高めている

```
1 import java.util.Scanner;
2 public class NotObject {
3     int syutudai;           //出題する問題の添字
4     int seikaisuu = 0;     //正解数
5     String[] mondai = new String[3];
6     String[] kotae = new String[3];
7     //問題をセットするメソッド
8     public void mondaiSet(){
9         mondai[0] = "犬を英語で言うと?";
10        mondai[1] = "10進数の15を2進数4桁にすると?";
11        mondai[2] = "1秒間に転送できるデータのビット数の単位は?";
12        kotae[0] = "dog";
13        kotae[1] = "1111";
14        kotae[2] = "bps";
15    }
16    //問題を出題するメソッド
17    public void syutudai(){
18        syutudai = (int) (Math.random() * 3);
19        System.out.println( _____ );
20    }
21    //正解かどうか判定するメソッド
22    public String hantei(String kaitou){
23        if(kotae[syutudai].equals(kaitou)){
24            seikaisuu++;
25            _____ "正解です!";
26        }else{
27            _____ "不正解です";
28        }
29    }
30    //メインメソッド
31    public static void main(String[] args) {
32        String kaitou;           //プレイヤーが回答した答え
33        Scanner sc = new Scanner(System.in);
34        NotObject no = new NotObject(); //自分自身をインスタンス化
35        _____ ;
36        //5問出題する繰り返し処理
37        for(int kaisu = 1; kaisu < 6; kaisu++){
38            System.out.printf("***** 第 %d 問 *****\n", kaisu);
39            _____ ;
40            kaitou = sc.next();
41            System.out.println(no.hantei(kaitou));
42        }
43        sc.close();
44        System.out.printf("正解数は %d 問です", _____ );
45    }
46 }
```

オブジェクト指向で処理するためには・・・

参考文献 小森裕介「なぜ、あなたは Java でオブジェクト指向開発ができないのか」技術評論社、2005 年

オブジェクト指向により分析を行う（モデリング）

- ①プログラムの仕様を決める
- ②クラスを抽出する
- ③オブジェクト間のメッセージのやり取りを考える
- ④各クラスのメソッドを洗い出す
- ⑤各クラスの属性（フィールド）を洗い出す

※クラスを抽出する際のヒント …… プログラムの仕様から「名詞」を抜き取る

※本来は、このモデリングで UML（クラス図やシーケンス図など）を作成する

※どのようなクラスにするか、どのクラスにメソッドを実装するか、というのはモデリング次第
同じ仕様でもオブジェクト指向に落とし込む段階で様々な答えが出てくる。

現実世界をオブジェクト指向に落とし込む、このモデリングという作業を経ないとオブジェクト指向型プログラミングを会得する事はできない。
つまり、最も大切なのはこのモデリングの作業と言える。しかし、これは非常に難しいだけでなく慣れも必要である。
授業や検定と関係あるかどうかに関わらず、このモデリングが OOP の肝である。

*** どのようなクラスが必要か、どのようなメソッドが必要か、メモ書きしてみよう ***

プログラム3 オブジェクト指向で処理 (OOP)

Mondai クラス

```
1 public class Mondai {
2     private String mondaibun;    //問題文
3     private String kotae;       //解答
4     //コンストラクタ
5     public Mondai(String mondaibun, String kotae){
6         this.mondaibun = mondaibun;
7         this.kotae = kotae;
8     }
9     //ゲッターメソッド
10    public String getMondaibun(){
11        return mondaibun;
12    }
13    public String getKotae(){
14        return kotae;
15    }
16 }
```

Player クラス

```
1 import java.util.Scanner;
2 public class Player {
3     private int seikaisuu = 0;
4     //回答する
5     public String kaitou(){
6         Scanner sc = new Scanner(System.in);
7         String kaitou = sc.next();
8         sc.close();
9         _____;
10    }
11    //正解かどうかの判定を先生から聞く
12    public void hanteiKiku(boolean hanteiKekka){
13        if( _____ ){
14            seikaisuu++;
15        }
16    }
17    //ゲッターメソッド
18    public int getSeikaisuu(){
19        return seikaisuu;
20    }
21 }
```

Sensei クラス

```
1 public class Sensei {
2     private Mondai[] mondai = new Mondai[3];
3     //コンストラクタで問題を用意する
4     public Sensei(){
5         _____;
6     }
7     //問題を用意する
8     private void mondaiSet(){
9         mondai[0] = new Mondai("犬を英語で言うと?", "dog");
10        mondai[1] = new Mondai("10進数の15を2進数4桁にすると?", "1111");
11        mondai[2] = new Mondai("1秒間に転送できるデータのビット数の単位は?", "bps");
12    }
13    public void startQuiz(Player player){
14        //クイズを開始する
15        for(int kaisu = 1; kaisu < 6; kaisu++){
16            //出題する
17            System.out.printf("***** 第 %d 問 *****\n", kaisu);
18            int syutudai = (int) (Math.random() * 3);
19            System.out.println( _____ );
20            //プレイヤーから回答を聞く
21            String kaitou = _____;
22            //正解かどうか判定する
23            if(mondai[syutudai].getKotae().equals(kaitou)){
24                System.out.println("正解です!");
25                player.hanteiKiku(true); //プレイヤーに結果を返す
26            }else{
27                System.out.println("不正解です");
28                player.hanteiKiku(false); //プレイヤーに結果を返す
29            }
30        }
31    }
32 }
33
```

メインメソッドを含むクラス

```
1 public class Main {
2     public static void main(String[] args) {
3         Player player = new Player(); //プレイヤーを用意
4         Sensei sensei = new Sensei(); //出題者を用意
5         _____; //クイズをスタート
6         System.out.printf("正解数は %d 問です", _____ );
7     }
8 }
```

特別プログラム クイズゲームを通してOOPを学ぶ

処理内容

問題を出題し、回答をキーボードから入力する。問題は全部で3問で、出題回数は5回とする。5回出題し終わったら、正解数を表示してプログラムを終了する。

※本来は、3問しかない問題を5回の中で出題するのは不自然だが、プログラムを簡略化するためそのような処理とした。

実行結果

```
***** 第 1 問 *****
10進数の15を2進数4桁にすると？
1101
不正解です
***** 第 2 問 *****
犬を英語で言うと？
dog
正解です！
***** 第 3 問 *****
犬を英語で言うと？
dog
正解です！
***** 第 4 問 *****
犬を英語で言うと？
dog
正解です！
***** 第 5 問 *****
1秒間に転送できるデータのビット数の単位は？
pps
不正解です
正解数は 3 問です
```

プログラム1 全てメインメソッドで処理 (OOPではない)

```
1 import java.util.Scanner;
2 public class NotObject2 {
3     public static void main(String[] args) {
4         int seikaisuu = 0;           //正解数
5         int syutudai;               //出題する問題の添字
6         String kaitou;              //プレイヤーが回答した答え
7         String[] mondai = new String[3];
8         String[] kotae = new String[3];
9         mondai[0] = "犬を英語で言うと?";
10        mondai[1] = "10進数の15を2進数4桁にすると?";
11        mondai[2] = "1秒間に転送できるデータのビット数の単位は?";
12        kotae[0] = "dog";
13        kotae[1] = "1111";
14        kotae[2] = "bps";
15        Scanner sc = new Scanner(System.in);
16        //5問出題する繰り返し処理
17        for(int kaisu = 1; kaisu < 6; kaisu++){
18            System.out.printf("***** 第 %d 問 *****\n", kaisu);
19            syutudai = (int) (Math.random() * 3);
20            System.out.println(mondai[syutudai]);
21            kaitou = sc.next();
22            if(kotae[syutudai].equals(kaitou)){
23                System.out.println("正解です!");
24                seikaisuu++;
25            }else{
26                System.out.println("不正解です");
27            }
28        }
29        sc.close();
30        System.out.printf("正解数は %d 問です", seikaisuu);
31    }
32 }
```


プログラム2 メインメソッドとインスタンスメソッドで処理 (OOPではない)

※全商検定2級のような形、サブルーチン化して保守性や可読性を高めている

```
1 import java.util.Scanner;
2 public class NotObject {
3     int syutudai;           //出題する問題の添字
4     int seikaisuu = 0;     //正解数
5     String[] mondai = new String[3];
6     String[] kotae = new String[3];
7     //問題をセットするメソッド
8     public void mondaiSet(){
9         mondai[0] = "犬を英語で言うと?";
10        mondai[1] = "10進数の15を2進数4桁にすると?";
11        mondai[2] = "1秒間に転送できるデータのビット数の単位は?";
12        kotae[0] = "dog";
13        kotae[1] = "1111";
14        kotae[2] = "bps";
15    }
16    //問題を出題するメソッド
17    public void syutudai(){
18        syutudai = (int) (Math.random() * 3);
19        System.out.println(mondai[syutudai]);
20    }
21    //正解かどうか判定するメソッド
22    public String hantei(String kaitou){
23        if(kotae[syutudai].equals(kaitou)){
24            seikaisuu++;
25            return "正解です!";
26        }else{
27            return "不正解です";
28        }
29    }
30    //メインメソッド
31    public static void main(String[] args) {
32        String kaitou;       //プレイヤーが回答した答え
33        Scanner sc = new Scanner(System.in);
34        NotObject no = new NotObject(); //自分自身をインスタンス化
35        no.mondaiSet();
36        //5問出題する繰り返し処理
37        for(int kaisu = 1; kaisu < 6; kaisu++){
38            System.out.printf("***** 第 %d 問 *****\n", kaisu);
39            no.syutudai();
40            kaitou = sc.next();
41            System.out.println(no.hantei(kaitou));
42        }
43        sc.close();
44        System.out.printf("正解数は %d 問です", no.seikaisuu);
45    }
46 }
```

オブジェクト指向で処理するためには・・・

小森裕介「なぜ、あなたは Java でオブジェクト指向開発ができないのか」技術評論社、2005 年 参照

オブジェクト指向により分析を行う（モデリング）

- ①プログラムの仕様を決める
- ②クラスを抽出する
- ③オブジェクト間のメッセージのやり取りを考える
- ④各クラスのメソッドを洗い出す
- ⑤各クラスの属性（フィールド）を洗い出す

※クラスを抽出する際のヒント …… プログラムの仕様から「名詞」を抜き取る

※本来は、このモデリングで UML（クラス図やシーケンス図など）を作成する

※どのようなクラスにするか、どのクラスにメソッドを実装するか、というのはモデリング次第
同じ仕様でもオブジェクト指向に落とし込む段階で様々な答えが出てくる。

**現実世界をオブジェクト指向に落とし込む、このモデリングという作業を経ないとオブジェクト指向型プログラミングを会得する事はできない。
つまり、最も大切なのはこのモデリングの作業と言える。しかし、これは非常に難しいだけでなく慣れも必要である。
授業や検定と関係あるかどうかに関わらず、このモデリングが OOP の肝である。**

*** どのようなクラスが必要か、どのようなメソッドが必要か、メモ書きしてみよう ***

プログラム3 オブジェクト指向で処理 (OOP)

Mondai クラス

```
1 public class Mondai {
2     private String mondaibun;    //問題文
3     private String kotae;        //解答
4     //コンストラクタ
5     public Mondai(String mondaibun, String kotae){
6         this.mondaibun = mondaibun;
7         this.kotae = kotae;
8     }
9     //ゲッターメソッド
10    public String getMondaibun(){
11        return mondaibun;
12    }
13    public String getKotae(){
14        return kotae;
15    }
16 }
```

Player クラス

```
1 import java.util.Scanner;
2 public class Player {
3     private int seikaisuu = 0;
4     //回答する
5     public String kaitou(){
6         Scanner sc = new Scanner(System.in);
7         String kaitou = sc.next();
8         sc.close();
9         return kaitou;
10    }
11    //正解かどうかの判定を先生から聞く
12    public void hanteiKiku(boolean hanteiKekka){
13        if(hanteiKekka){
14            seikaisuu++;
15        }
16    }
17    //ゲッターメソッド
18    public int getSeikaisuu(){
19        return seikaisuu;
20    }
21 }
```

Sensei クラス

```
1 public class Sensei {
2     private Mondai[] mondai = new Mondai[3];
3     //コンストラクタで問題を用意する
4     public Sensei(){
5         mondaiSet();
6     }
7     //問題を用意する
8     private void mondaiSet(){
9         mondai[0] = new Mondai("犬を英語で言うと?", "dog");
10        mondai[1] = new Mondai("10進数の15を2進数4桁にすると?", "1111");
11        mondai[2] = new Mondai("1秒間に転送できるデータのビット数の単位は?", "bps");
12    }
13    public void startQuiz(Player player){
14        //クイズを開始する
15        for(int kaisu = 1; kaisu < 6; kaisu++){
16            //出題する
17            System.out.printf("***** 第 %d 問 *****\n", kaisu);
18            int syutudai = (int) (Math.random() * 3);
19            System.out.println(mondai[syutudai].getMondaibun());
20            //プレイヤーから回答を聞く
21            String kaitou = player.kaitou();
22            //正解かどうか判定する
23            if(mondai[syutudai].getKotae().equals(kaitou)){
24                System.out.println("正解です!");
25                player.hanteiKiku(true); //プレイヤーに結果を返す
26            }else{
27                System.out.println("不正解です");
28                player.hanteiKiku(false); //プレイヤーに結果を返す
29            }
30        }
31    }
32 }
33
```

メインメソッドを含むクラス

```
1 public class Main {
2     public static void main(String[] args) {
3         Player player = new Player(); //プレイヤーを用意
4         Sensei sensei = new Sensei(); //出題者を用意
5         sensei.startQuiz(player); //クイズをスタート
6         System.out.printf("正解数は %d 問です", player.getSeikaisuu());
7     }
8 }
```