

オブジェクト指向プログラミングII

第3回 リファクタリングの演習
10月10日



本日の予定

- ▶ 金種計算のプログラムのリファクタリング
 - ▶ Scombで質問に回答しながらリファクタリング
- ▶ リファクタリングの解説のつづき

リファクタリングとは

外部から見た時の振る舞いを保ちつつ、
理解や修正が簡単になるようにソフトウェアの内部構造を変化させること



例題

- ▶ はじめは「ある特定の状況の計算を行い、結果を確認できれば良い」と考えてプログラムを作成。
 - ▶ 10円硬貨4枚, 100円硬貨6枚, 500円硬貨2枚を全部または一部を使って支払える金額は何通りか.

本日のリファクタリングのキーワード

- ▶ シンボリック定数によるマジックナンバーの置き換え
 - ▶ 使われている数値の意味は？
- ▶ 制御フラグの削除
 - ▶ あるケースにはフラグを立てておくことで、つぎの処理を制御する
- ▶ メソッドの抽出
 - ▶ 手続きに意味のある名前をつけていろいろな場所で利用できるようにする

```

public class Example{
    static public void main(String[] args){
        int x,y,z,k=0,i,ksum,not;
        int[] isum = new int[106];
        for(x=0; x<=4; x++){
            for(y=0; y<=6; y++){
                for(z=0; z<=2; z++){
                    ksum=10*x+100*y+500*z;
                    /* check if previous sum equals to ksum */
                    not=0;
                    for(i=1; i<=k; i++){
                        if(isum[i] == ksum) not=1;
                    }
                    if(not == 0){
                        k+=1; isum[k]=ksum;
                        /* System.out.println("k= " + k + ", ksum= " + ksum +
                            ", x = " + x + ", y = " + y + ", z = " + z); */
                    }else{
                        System.out.println("not k= " + k + ", x = " + x + ", y = " + y + ", z = " + z);
                    }
                }
            }
        }
        System.out.println("k= " + k);
    }
}

```

Javaで書き換えそのまま
これを実行



```
not k= 15, x = 0, y = 5, z = 0
not k= 15, x = 0, y = 5, z = 1
not k= 16, x = 0, y = 6, z = 0
not k= 16, x = 0, y = 6, z = 1
not k= 32, x = 1, y = 5, z = 0
not k= 32, x = 1, y = 5, z = 1
not k= 33, x = 1, y = 6, z = 0
not k= 33, x = 1, y = 6, z = 1
not k= 49, x = 2, y = 5, z = 0
not k= 49, x = 2, y = 5, z = 1
not k= 50, x = 2, y = 6, z = 0
not k= 50, x = 2, y = 6, z = 1
not k= 66, x = 3, y = 5, z = 0
not k= 66, x = 3, y = 5, z = 1
not k= 67, x = 3, y = 6, z = 0
not k= 67, x = 3, y = 6, z = 1
not k= 83, x = 4, y = 5, z = 0
not k= 83, x = 4, y = 5, z = 1
not k= 84, x = 4, y = 6, z = 0
not k= 84, x = 4, y = 6, z = 1
k= 85
```

実行結果を確認



Q.変数の意味を答えなさい

```
int x
int y
int z
int k=0
int i
int ksum
int not
int[] isum
```



Q.変数宣言の位置を変更できる場所は
変更しなさい

ヒント: 繰り返しを制御する変数





Q.数字の意味を答えなさい

106

0

4

6

2

10

500

100

1



Q.分かりにくい数字を定数で置き換えなさい



```

public class Example{
    static public void main(String[] args){
        int ksum;
        int not;
        int[] isum = new int[106];
        int k =0;
        for(int x=0; x<=4; x++){
            for(int y=0; y<=6; y++){
                for(int z=0; z<=2; z++){
                    ksum=10*x+100*y+500*z;
                    /* check if previous sum equals to ksum */
                    not=0;
                    for(int i=1; i<=k; i++){
                        if(isum[i] == ksum) not=1;
                    }
                    if(not == 0){
                        k+=1; isum[k]=ksum;
                        System.out.println("k= " + k + ", ksum= " + ksum +
                            ", x = " + x + ", y = " + y + ", z = " + z);
                    }else{
                        System.out.println("not k= " + k + ", x = " + x + ", y = " + y + ", z = " + z);
                    }
                }
            }
        }
        System.out.println("k = " + k);
    }
}

```



Q. 点線部分に着目し
変数 not の使い方を説明しなさい



Q. 点線部分で行っている処理を説明しなさい



制御フラグ



- ▶ フラグとは、flag(旗)のことで、「データにつけくわえられて、それについての情報を与える追加部分」(コンピュータ用語辞典; Blue Backs)
- ▶ プログラミングの際にある条件が成立したかどうかを判断するために使われる変数。成立しているときは1、成立していないときは0というような使い方をする。(日経パソコン用語事典)

Q. 制御フラグの除去



- ▶ 組み合わせで生成された金額がすでに存在する金額かを調べるメソッド

`boolean isContain(int money);`

を定義し、変数`not`を除去して、プログラムを書き換えなさい。

- ▶ 書き換えが終了したら、コンパイル、実行し、結果が正しいことを確認しなさい。



Q. メソッドの抽出

main メソッドから
金種を計算するメソッド

```
void calculate ();
```

と、支払える金額と何通りあるかの結果を画面表示するメソッド

```
void display();
```

を定義し、プログラムを書き換えなさい。

mainメソッドは

```
static public void main(String[] args){
```

```
    calculate();
```

```
    display();
```

```
}
```

となる。

- ▶ 書き換えが終了したら、コンパイル、実行し、結果が正しいことを確認しなさい。

```
1:0
2:500
3:1000
4:100
5:600
6:1100
7:200
8:700
9:1200
10:300
11:800
12:1300
...
80:1340
81:440
82:940
83:1440
84:1540
85:1640
金額の種類は85通り
```