

Classの設計の基本

クラスを利用してデータを隠蔽化する。
データを変更するには決められたメソッドを通じてのみ変更できる。

```
import java.io.*; // クラス設計の基本
public class DataClass {
    private int data; //このクラスが管理したい値

    //通常は、1つの変数に対して、セッターとゲッターをペアで準備する
    public void setData( int x){ data = x; } //値を代入するためのセッター
    public int  getData(){ return (data); } //値を得るためのゲッター
}
```

変数はprivate指定で隠蔽化

上記のクラスを利用するメインクラス

```
import java.io.*;
public class DataMain {
    public static void main(String args[]) {
        DataClass mydata = new DataClass();
        mydata.setData(100);
        System.out.println( mydata.getData() );
    }
}
```

メソッドはpublic指定で公開
(公開することでmainメソッドから、利用できるようにする。この他にprotectedがある)

セッターの名前は set変数名(1文字目は大文字)
ゲッターの名前は get変数名(1文字目は大文字)

Classの設計の基本演習

リモコンのコントローラを例にクラスを設計します。リモコンの内部

```
public class RC {
    private int on; //このクラスが管理したい値

    //通常は、1つの変数に対して、セッターとゲッターをペアで準備する
    public void setOn( int x){ on = x; } //値を代入するためのセッター
    public int  getOn(){ return (on); } //値を得るためのゲッター
}
```

```
import java.io.*;
public class RCMain {
    public static void main(String args[]) {
        RC myRC = new RC();
        System.out.println( "スイッチは:"+myRC.getOn() ); //0:スイッチが入っていない
        myRC.setOn( 1 ); //スイッチを入れる(1にする)
        System.out.println( "スイッチは:"+myRC.getOn() ); //1:スイッチが入っている
        myRC.setOn( ここを考える ); //スイッチをOFFにする(0にする)
        System.out.println( "スイッチは:"+ここを考える ); //0:スイッチが入っていない
    }
}
```



課題9

下記を参考に一度に2つのサイコロを同時に振って表示させる

```
import java.math.*;
class Dice {
    public void play () {
        int randomNumber = ( (int) Math.random() ) % 6 +1 ;
        System.out.println("Dice = [" +randomNumber+ "]");
    }
}
```

ファイル名 : Dice.java

```
import java.io.*;
public class DiceMain {
    public static void main(String args[]) {
        Dice mydice= new Dice();
        mydice.play();
    }
}
```

ファイル名 : DiceMain.java

課題10

おみくじゲームを、2つのクラスを使って作成する。

```
import java.math.*;
class OmikujiClass {
    public void play() {
        double randomNumber = Math.random();
        String result = "";
        if (randomNumber < 0.2) { result = "凶です"; }
        ....ここら辺は自分で考える

        System.out.println(result);
    }
}
```

ファイル名 : OmikujiClass.java

```
class OmikujiMain{
    public static void main (String args[]) {
        OmikujiClass myOmikuji = new OmikujiClass();
        ....ここら辺は自分で考える
    }
}
```

ファイル名 : OmikujiMain.java