Java講習　第8回

～クラス　その１～

Java：　同じ処理はまとめる！

→　メソッド・クラス　がある。

Javaのプログラムは必ず含まれてる。

メソッド とは

- 処理の塊で、関数のようなもの

例

y=f(x)=x^2+3x+5
x=5の時のyを
計算したい！

y=f(5);

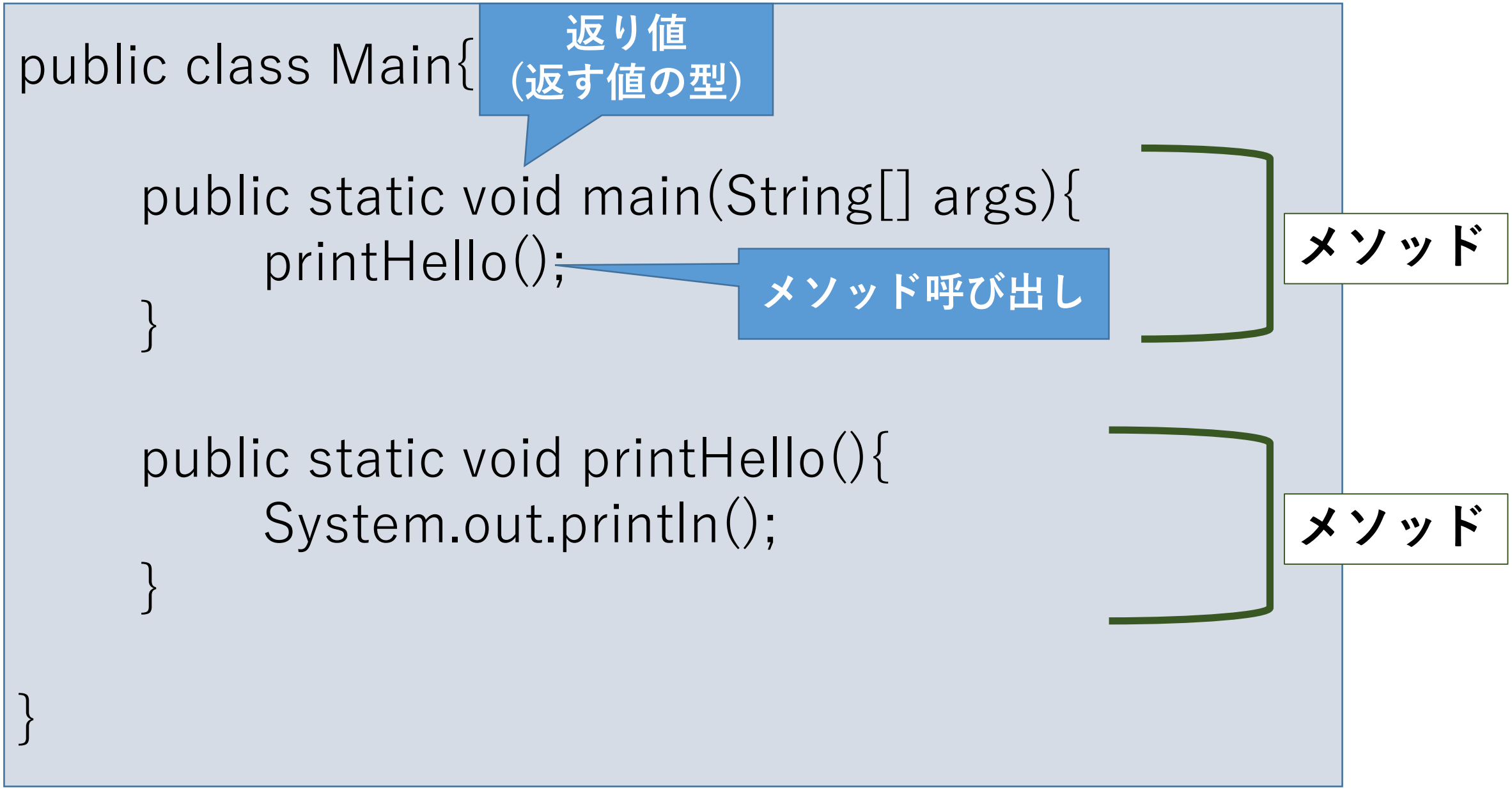
メソッド

与えられた数字5
をxとして
X^2+3x+5を計
算

```
int fanction(int x){
    return x*x+3*x+5;
}
```

計算された答えy
がもらえる

y=f(5);


public class Main{
返り値
(返す値の型)
public static void main(String[] args){
printHello();
メソッド呼び出し
}
メソッド
public static void printHello(){
System.out.println();
}
メソッド
}

クラスとは…

- プログラムの一番外側に記述されていたもの

```
public△class△Main{
        public△static△void△main(String[]△args){
                System.out.println(“Hello world!”);
        }
}
```

2つ以上あっても問題なし!

```
public△class△MainClass{
    public△static△void△main(String[]△args){
        System.out.println(“Hello world!”);
    }
}
class△Sub1Class{
    public void subPrint(){
        System.out.println(“subです。”);
    }
}
```

クラスのつくり方

```
class△クラス名{
}
```

```
class△クラス名{
        int a;
        void method(){
        }
}
```

クラスのつくり方

ファイル名
(MainClass.java)

```
public△class△MainClass{
    public△static△void△main(String[]△args){
        System.out.println(“Hello world!”);
    }
}
class△Sub1Class{
    public void subPrint(){
        System.out.println(“subです。”);
    }
}
```

好きなクラス名

クラスのつくり方

```
public△class△MainClass{
        public△static△void△main(String[]△args){
                System.out.println(“Hello world!”);
        }
}
class△Sub1Class{
        void subPrint(){
                System.out.println(“subです。”);
        }
}
```

※ファイル名と違うクラスを作るときはクラスにPublicを付けない

クラスとは...

- **メソッド**と**変数**の集まり

```
public△class△Car{
        int   speed;
        int   size;
        String   color;
        void   accel(){
        }
        void   brake(){
        }
}
```

クラスとは…

- **テンプレートのような雛型，設計書，定義⇒複製可能**

```
public△class△Car{
        int   speed;
        int   size;
        String   color;
        void   accel(){
        }
        void   brake(){
        }
}
```

値を変えると種類が変わる
骨格は同じ

```
public△class△Car{
        int   speed =40;
        int   size = 300;
        String   color;
        void   accel(){
        }
        void   brake(){
        }
}
```

```
public△class△Car{
        int   speed =120;
        int   size = 100;
        String   color;
        void   accel(){
        }
        void   brake(){
        }
}
```

クラスはあくまでテンプレート(型)

- 中身が詰まってない
- 中身が無いので機能（メソッド,変数）が使えない
- 中身を詰める必要がある

中身　　・・・**インスタンス**
中身を詰める・・・**インスタンスを生成**

インスタンス生成方法 (クラスの使い方)

- **new 演算子を使う**

クラス名△変数　＝ new△クラス名();

インスタンス生成

呼ぶクラス専用の
型を用意

インスタンス
を変数で保持

インスタンス生成方法 (クラスの使い方)

クラス名△変数　＝ new△クラス名();

例:

Subクラスを呼ぶ場合

Sub△sub1 ＝ new Sub();

Sub△sub2 ＝ new Sub();

クラス使用例

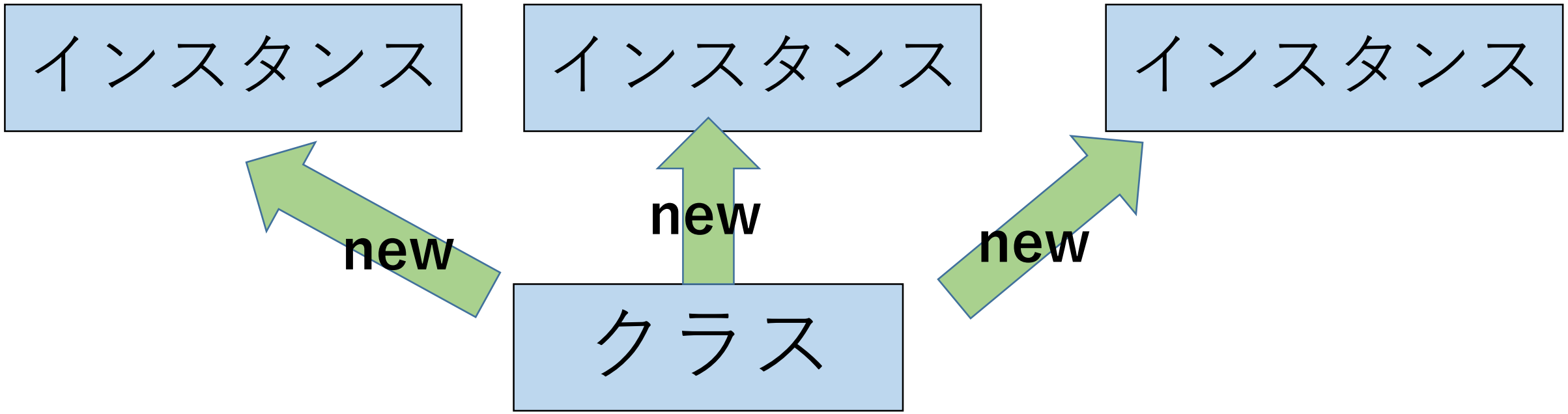
```
public class MainClass{
        public static void main(String[] args){
                Sub sub1 = new Sub();
        }
}
class Sub{
        void method1(){
        }
}
```

Sub クラス

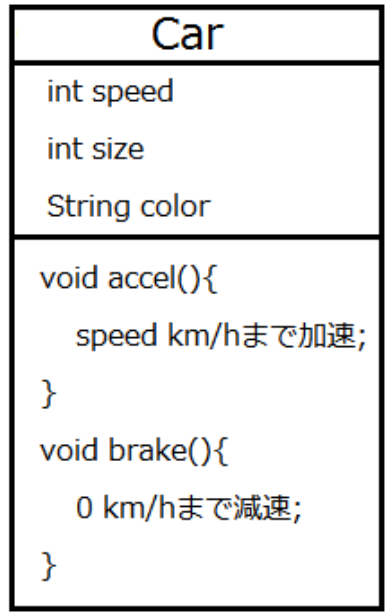
インスタンス生成すると…

- **インスタンスを生成したクラスのメソッド, 変数が使える!**
  →テンプレの機能が使える!

クラスを元に作られた操作できる部品



クラス

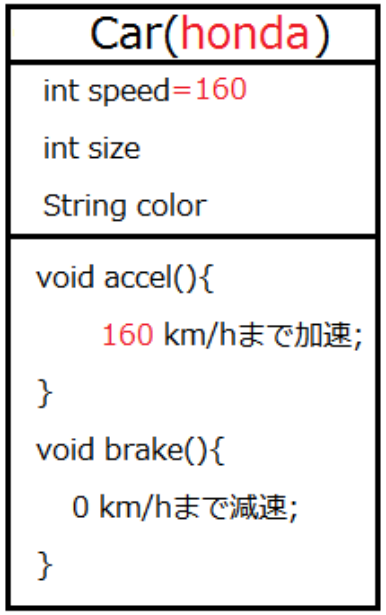


インスタンス生成

```
Car honda=new Car();
honda.speed=160;
```

インスタンス生成

```
Car toyota=new Car();
toyota.speed=180;
```



インスタンス生成後

インスタンス生成した
クラスの変数・メソッドの使い方

- **変数を使う**

  変数.クラス内の変数

※メソッド内の変数は呼べない

- **メソッドを使う**

  変数.メソッド名()

```
class Sub{
        int a;
        void method1(){
                int b;
        }
}
```

```
public class Main{
        public static void main(String[] args){
                Sub sub1 = new Sub();    //Subクラス型のsub1を宣言
                int i = sub1.i;    //sub1のiをMainclassのiに入れる
                        sub1.i = 20;              //sub1のiの値を変更
                sub1.method1();      //sub1のmethod1メソッドを呼び出し

        }
}
class Sub{
        int i=10;
        void method1(){
                System.out.println(i);
        }
}
```

Main クラス

Sub クラス

演習1

mainメソッドのあるクラスをAクラス
mainメソッドないクラスをBクラス
とここでは呼ぶ

(1)AクラスとBクラスを作成
(2)Bクラスに“Hello world2”と出力するメソッドを作る
(3)mainメソッドから(2)で作ったメソッドを呼び出す。

演習2

Aクラス、Televisionクラスを作る

Televisonクラスの仕様

変数:　int型のch メソッド1つ: chを出力する。

Aクラスから２つTelevisonクラスのインスタンスを生成
→変数chを変更

Televisonクラスのメソッドをそれぞれ呼ぶ