


# 10月～11月パート

## Java基礎編(説明)



# 学ぶ内容



- ・ クラス設計
- ・ インスタンス化

Java・DB編で必要となる知識となります。

前のパートのGitの使い方は大丈夫ですか??

次のパートでクラス設計の実践を学びます。今回は軽い説明。

# なぜクラス設計とインスタンス化を学ぶ？

C言語では、複数の関数を定義し使い分けることで役割分担をしていた。Javaでは、クラスというものを作成し役割分担をする。

C言語で関数を定義し使用する際、任意の場所で関数を呼び出す必要があった。Javaも同様に、クラスを使いたい場合、任意の場所でクラスを呼び出す必要がある。その動作をインスタンス化という。

C言語の場合

```
#include ...
```

```
int main(){
```

関数の呼び出し

```
int x = hogehoge();
```

```
return;
```

```
}
```

関数の定義

```
int hogehoge(){
```

```
}
```

Javaの場合

```
import ...
```

```
public class hogehoge{
```

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

```
ABC abc = new ABC();
```

インスタンス化

```
float x = 20.0;
```

```
abc.print(x);
```

```
}
```

```
}
```

クラス

```
public class ABC{
```

```
void print(float a){
```

```
System.out.println(a);
```

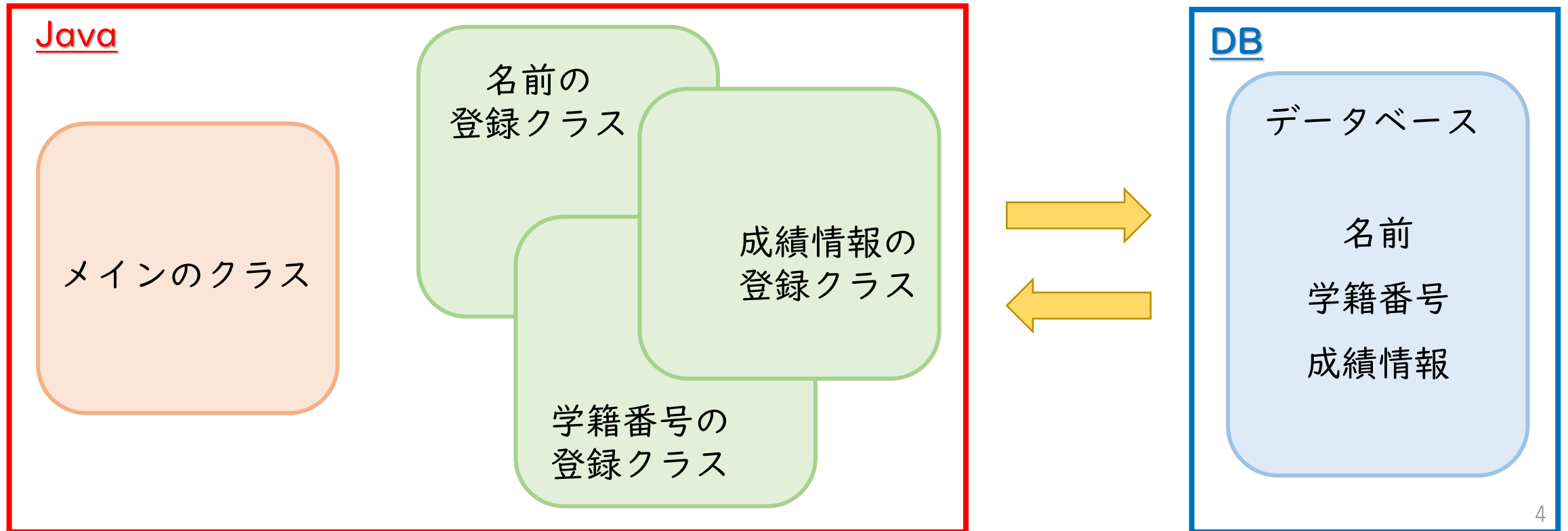
```
}
```

```
}
```

# なぜクラス設計とインスタンス化を学ぶ？

後のパートでDB(データベース)を学ぶのだが、そこでクラスとインスタンス化についての知識が必要となる。

Java基礎では、クラスとインスタンス化とは何かを理解してもらい、DBを導入した際にクラスとインスタンス化が重要であることを認知してもらうために説明・問題演習を通して学んでもらう。



# クラス設計とは？

クラス設計には、様々な方法がある。C言語では関数を定義した際に引数や戻り値を設定したと思う。クラス設計では、クラス自体には引数や戻り値を設定することはできない。(厳密には少し異なる) その代わり、クラス設計をする際にメソッドと呼ばれるC言語でいう関数を多数定義することができ、そのメソッドに引数や戻り値を設定することができる。(詳細は次回パート)

メインのクラスで任意のクラスをインスタンス化し、呼び出したクラスの中にあるメソッドをインスタンスを利用して呼び出しやりたい処理を行う。

## メインクラス

```
import ...
```

```
public class hogehoge{
    public static void main(String[] args){
        ABC abc = new ABC();
        DEF def = new DEF();
        def.aisatsu();
        System.out.print(abc.keisan(2));
    }
}
```

インスタンス化

インスタンス

メソッド呼び出し

## ABCクラス

```
public class ABC{
    void print(float a){
        System.out.println(a);
    }
    int keisan(int b){
        int y = b*b;
        return y;
    }
}
```

## DEFクラス

```
public class DEF{
    void aisatsu(void){
        System.out.println( “やあ” );
    }
}
```

メソッド

# インスタンス化とは？

インスタンス化とは、作成した新しいクラスを利用するための動作である。インスタンス化したクラスのメソッドの機能を集約したものをインスタンスという。特定のメソッドを呼び出したいときにインスタンスを利用してメソッドを呼び出す。(詳細は次次回パート)

```
import ...
```

```
public class hoge hoge {  
    public static void main(String[] args) {  
        ABC abc = new ABC(); インスタンス化  
        System.out.print(abc.keisan(2);); メソッド呼び出し  
    }  
}
```

abc.〇〇()とすることで、ABCクラスの〇〇メソッドを呼び出している。

ABCクラスには2つのメソッドがあり、abcというインスタンスには、printメソッドとkeisanメソッドの機能がある

```
public class ABC {
```

```
    void print(float a) {  
        System.out.println(a);  
    }
```

```
    int keisan(int b) {  
        int y = b*b;  
        return y;  
    }
```

メソッド

お疲れさまでした



次回は、クラス設計の実践です。