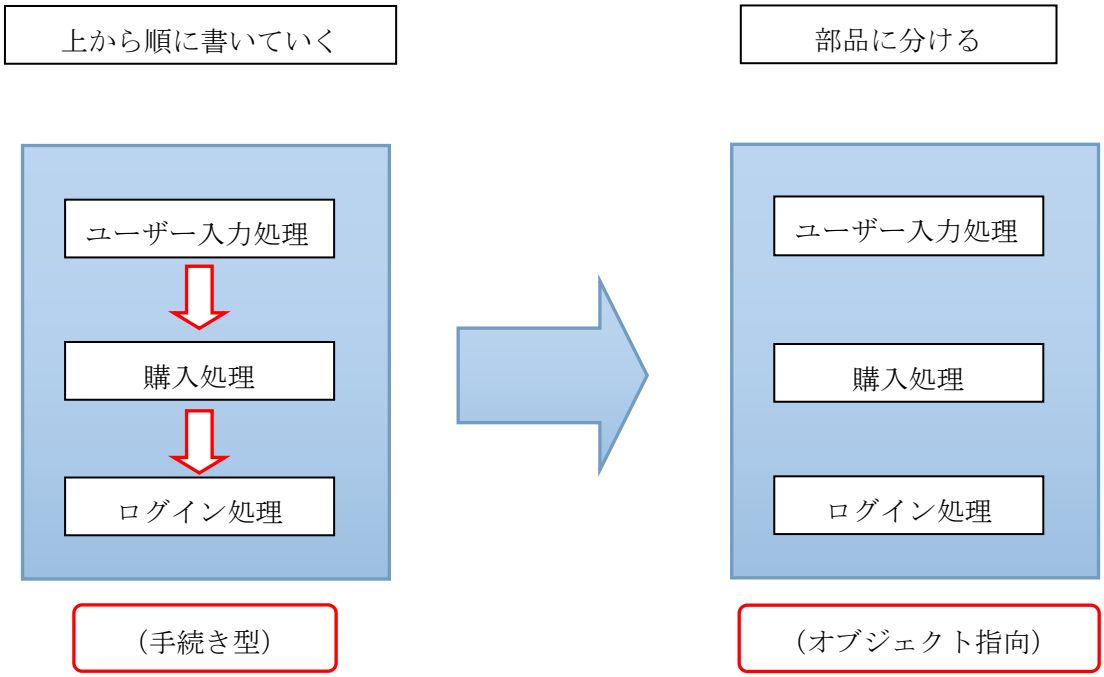
Java オブジェクト指向

オブジェクト：人、もの、概念
人、ものや考え方（概念）を別々にプログラムし、各オブジェクトを繋ぎ合わせ開発する考え方。”各オブジェクトがどこを向いているか”を考えるので「オブジェクト思考」ではなく「オブジェクト指向」という漢字を使います。
オブジェクト指向はかなり奥が深く、研修中で100%理解するのは難しいです。
テキストに載っている最低限の事だけでも理解をして頂ければ就職には問題ないです。また、参考書を読んでも本によって例え方や表現方法が違うため、余計に混乱してしまう可能性がありますので、オブジェクト指向の入りとして、このテキストを理解していきましょう。

メリット
・プログラムを容易に変更しやすくなる
・プログラムの一部を簡単に転用できるなど

例えばＥＣサイトを作る時



一つのプログラムを複数の部品に分けて作ることができます。
そうすることで・・・
・以前作ったログイン機能を新しいサイトで利用できる。
・プログラムの変更や削除が簡単にできるようになる。

オブジェクト指向の重要キーワードで以下のことを学習します。
詳しくは後のテキストで解説をするので、今はなんとなくのイメージを持ってください。

① インスタンス化・・・プログラムをコピーをするイメージ。世の中には既に完成した Java のプログラムがあり、それらをコピーして使うことができます。

② コンストラクタ・・・インスタンス化する時の窓口、受皿のイメージ。色々な形でコピーすることが出来ます。

③ カプセル化・・・隠しておきたい値（データ）の隠蔽をすること。そのなかで、必要な情報を書き込んだり取得したりすることが出来きます。

④ 継承・・・機能を拡張すること。既に出来上がっているものに新しい機能を付け加えることができます。

⑤ 実装・・・ルール作り。共通の機能を持ったプログラミングでの共通の部分をルール化してこのルールにもとづいてプログラミングを書いていくことです。

＜インスタンス化＞

インスタンス化はプログラムのコピーをするというイメージです。原本を作成しておいて、必要な項目だけを編集して複数作成をしていくことです。まずはどうやってインスタンス化をされていくのか確認しましょう。

例

String A = new String(“Hello”)

new が出てきたらインスタンス化です。ここでは文字列 Hello を「インスタンス化（コピー）したものを変数 A に代入しました。」

演習①

Java プロジェクト「Person」を作成しましょう。

「Person」クラスを作成して下記をプログラミングしましょう。

「Test」クラスを作成して下記プログラミングしましょう。

実行してみましょう。

```
public class Person{
    public String name = null;
    public int age = 0;
}
```

ここでは初期値として、それぞれ null と 0 を入れておきます。このことを「初期化」と言います。この中に Test クラスからの name と age を持ってきています。

main メソッドがある Test クラスを実行します。

```
public class Test{
    public static void main(String[] args){
Person taro = new Person();
taro.name=” 山田太郎”;
taro.age=20;

System.out.println(taro.name);
System.out.println(taro.age);
    }
```

Person taro=new Person()の new はインスタンス化するという意味。new が出てきたらインスタンス化と覚えてください。

taro は変数名（わかりやすい名前で）

右辺の Perosn()をインスタンス化して変数 taro に代入します。

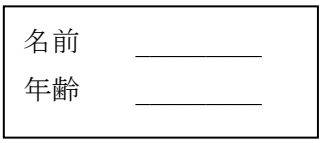
左辺の Person は型です。

taro.name の「.」を使う事によって taro インスタンスの name を呼び出す事が出来ます。

最後に taro.name と taro.age を出力するので山田太郎と２０が表示されます。

上記のプログラムを日常に例えてみます。
例えば、自分が先生だとして生徒さんの名前と年齢を確認したい！と思った時に、
一人ひとりに直接、名前と年齢を確認することも出来ますが、それは大変です。
そのような場合は、下記のように白紙に名前と年齢の欄だけを作ります。それを印刷して
全員に配ってからそれぞれ生徒さんに書いてもらった方が早いです。

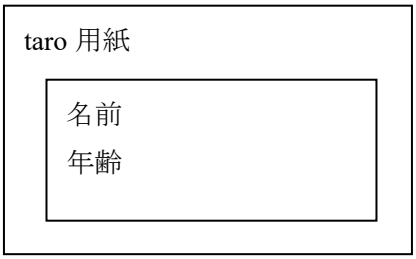
名前と年齢の欄だけの原本↓



プログラムで書くとこんな感じ↓

```
public class Person{
    public String name = null;
    public int age = 0;
}
```

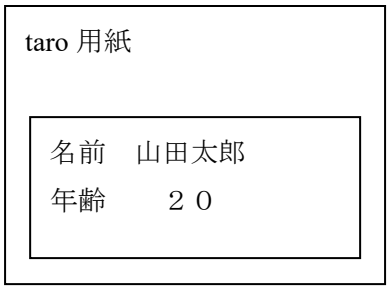
Test クラスの Person taro=new Person()というインスタンス化の記述は、原本である Person クラスのコピー（インスタンス）の名前を taro と設定するイメージです。




```
public class Test{
    public static void main(String[] args){
Person taro = new Person();
taro.name=”山田太郎”;
taro.age=20;
    }
}
```

taro.name と taro.age に対して、名前に山田太郎、年齢に 20 をそれぞれ代入します。
そして最後に、System.out.println(taro.name) System.out.println(taro.age)で出力します。

用紙の名前と年齢の欄が埋まる。



プログラムだとこんな感じ↓

```
public class Test{
    public static void main(String[] args){
Person taro = new Person();
taro.name=”山田太郎”;
taro.age=20;

System.out.println(taro.name);
System.out.println(taro.age);
    }
}
```

今回は山田太郎、２０をインスタンス化しましたが、他にも次郎さんや花子さんなど追加をすることが出来ます。以下の演習をやってみましょう。

（演習②）
インスタンス化を使って、
木村次郎、18
鈴木花子、16
（自身のお名前）、（自身のご年齢）
を Test クラスにプログラミングしてみましょう。

（演習③）
Person クラスに
public String phoneNumber = null;
public String address = null;
を追加してみましょう。

（演習④）
演習③で Person クラスに追加した情報を使って、Test クラスにすべて表示できるようプログラミングしてみましょう。また、これを表示してみましょう。

（演習⑤）
以下を参考に Person クラスにメソッドを追加してみましょう。

```
public class Person{
   public String name = null;
   public int age = 0;
       ・
       ・
       ・
   public void talk(){
      System.out.println(this.name + “が話す”);
   }
   public void walk(){
      System.out.println(this.name + “が歩く”);
```

```
    }
    public void run(){
        System.out.println(this.name + “が走る”);
    }
}
```

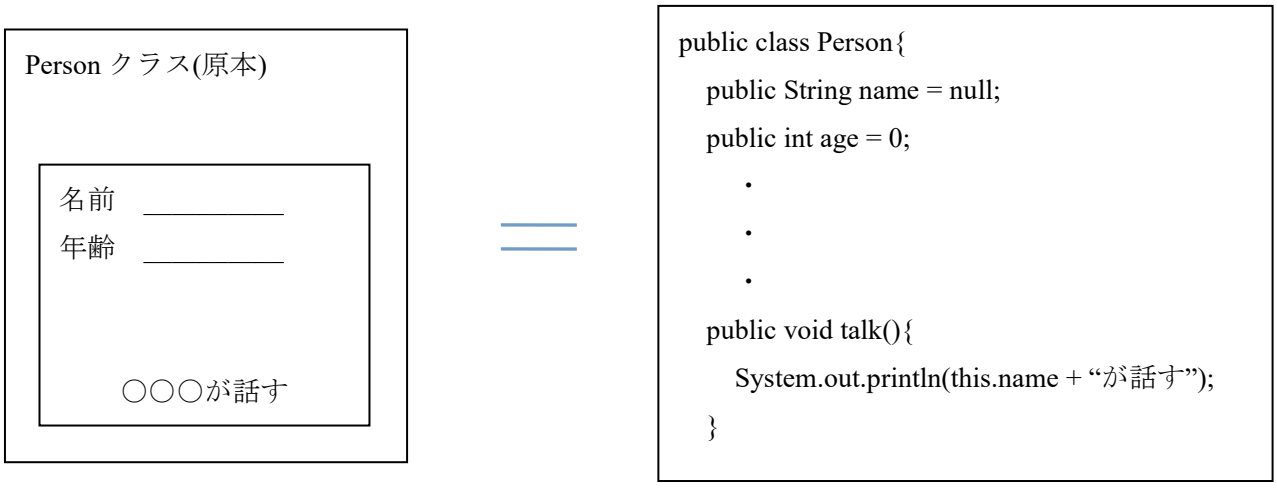
（演習⑥）

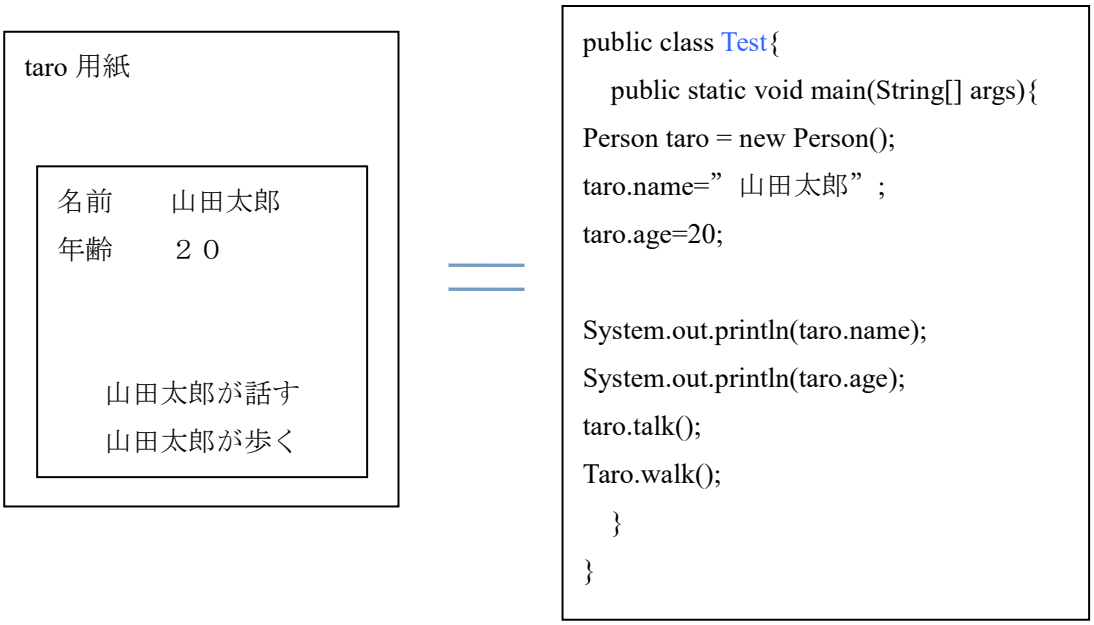
Test クラスにプログラムしたインスタンスを使って、演習⑤で追加したメソッドを呼び出してみましょう。

Test クラスでの呼び出し方

```
taro.talk();
taro.walk();
taro.run();
```

演習⑤と⑥を図でイメージすると・・・





今回は Person クラス（原本）にメソッドを書きましたので、新しく〇〇〇が話す、〇〇〇が歩く、という記述が追加されます。
「this.name」は、「このインスタンスの name」をさしています。
今回だと Person クラスのインスタンスである taro 内の「public String name;」を指しています。
Test クラスの「taro.name=”山田太郎”」により、taro インスタンスの name フィールドに「山田太郎」が代入されます。
Test クラスに書いた「taro.talk();」により、talk()メソッドの中身「System.out.println(this.name + “が話す”);」が呼びされ、結果「山田太郎が話す」ということになります。

（演習⑦）
あたらしく Robot クラスを作成してみましょう。
この中に、public String name = null; をプログラムしましょう。
また、talk()、walk()、run()メソッドをプログラムしましょう。

（演習⑧）
作成した Robot クラスを使って、Test クラスでインスタンス化してみましょう。
また、インスタンスの名前は、aibo、asimo、pepper としてプログラムしてみましょう。
完成したら、実行してみましょう。
