＜カプセル化＞
カプセル化とは、フィールドへの読み書きや、メソッドの呼び出しを制限する機能です。

例えば…

・このメソッドは、A クラスからは呼び出せるが、
　B クラスからは呼び出しができない。
・このフィールドの内容は、誰でも読めるけど書き換えは禁止。

といったことができます。

Java におけるカプセル化とは、大切な情報(フィールド)について、アクセス制御をかけることにより、悪意や間違いによる利用を停止、防止して、想定しない利用が発生したならば、その原因箇所を特定するための仕組みです。

<アクセス制御の範囲と指定方法>

制限の範囲	名前	記入方法	アクセスを許可する範囲
(制限が厳しい) ↕ (制限が緩い)	・ private ・ package private ・ protected ・ public	・ private ・(何も書かない) ・ protected ・ public	・自分のクラスのみ ・自分のパッケージのみ ・自分と同じパッケージか自分を継承したクラス ・すべてのクラス

名前の欄にある private や public などはアクセス修飾子とよばれ、フィールドやメソッドを宣言する際、先頭に記述することで、アクセス制御が可能になります。
Java には 4 種類のアクセス修飾子がありますが、この研修では private、public を覚えて頂ければ大丈夫です。

アクセス修飾子によって決まる「アクセスさせる範囲」の事を「スコープ」といいます。

（演習①）

Java プロジェクト「Capsule」を作成しましょう。

以下の Person クラスを作成後、プログラミングしましょう。

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

  public Person (String name,int age){
    this.name=name;
    this.age=age;
  }

  public String getName(){
    return this.name;
  }

  public void setName(String name){
    this.name = name;
  }

}
```

(String name, int age)の型でインスタンス化するためのコンストラクタを宣言します。

get の後の 1 文字は大文字になるので注意。
get の時は必ず return します。
今回だと getName でフィールドの値の取得をして呼び出し元に返します。

set の後の 1 文字も大文字になるので注意。
set の時は戻り値の型の位置に void と書きます。
今回だと setName で名前を書き換える「処理」を行うだけなので戻り値なしの void になります。

（演習②）

以下の Capsule クラスを作成後、プログラミングしましょう。

```
public class Capsule{
  public static void main(String[] args){
     Person taro = new Person(“山田太郎”,20);
     System.out.println(taro.name());
  }
}
```

（演習③）

Capsule プロジェクトを実行してみましょう。

（演習④）

Person クラスのフィールドの public を private に変更して隠蔽してみましょう。

```
public String name = null;
public int age = 0;
```

→

```
private String name = null;
private int age = 0;
```

private に変更すると、Capsule クラス System.out.println(taro.name)の name にエラーが出ます。private にした事によってアクセス許可の範囲が制限されてしまい、他のクラスから見えない状態になってしまい、カーソルを合わせると「不可視」というエラーメッセージが表示されます。
このような状態からカプセル化を使って、一部のデータを取り出したり、書き換えたりすることができます。
[例えば、役所には自分の個人情報が管理されてます。それを他の人全員に持っていかれてしまったり書き換えられてしまったら困りますよね。そこで、カプセル化をすることで、自分だけが欲しい情報を取り出したり、引っ越しで住所が変わる場合は住所変更をすることができます。]
ここで必要になってくるのが get(ゲッター)と set(セッター)というメソッドです。
get は値を取得すること。set は登録すること。

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

        public Person(String name,int age){
                this.name=name;
                this.age=age;
        }

        public String getName( ){
            return this.name;
        }

        public void setName(String name){
            this.name = name;
        }

}
```

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

        Person taro = new Person("山田太郎",20);

        System.out.println(taro.getName( ));

        taro.setName("花子");
        System.out.println(taro.getName( ));
        }
}
```

Capsule クラスの 1 つ目の青枠の中の赤文字を name から getName（get の後は大文字）に変更することで読み取る（name を get）ことが出来るようになります。また、Capsule クラスの赤枠を追加することによって、山田太郎という名前を花子へ書き換える（name を set）ことが出来ます。ただし setter は書き換え処理をするだけなので、書き換えた後の name（花子）を表示したければ、2 つ目の青枠のように再度 getter を利用し、書き換え後の name の値を読み取る必要があります。

（演習⑤）

getter を使って年齢を取得してみましょう。

setter を使って年齢を 30 歳に登録してみましょう。