

# プログラム設計

<http://bit.ly/design4d>

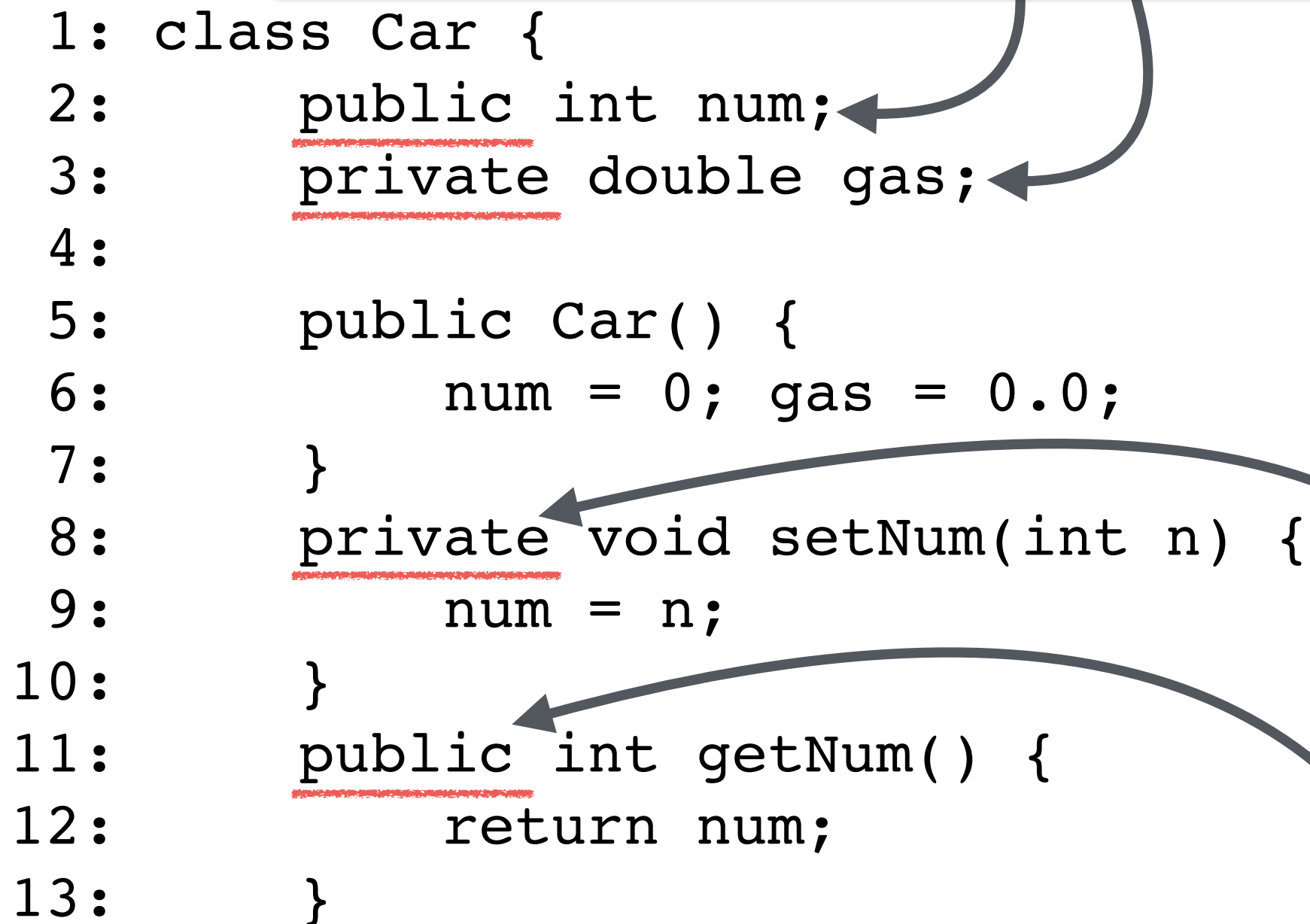
## 可視化によるメンバへのアクセス制御

前期 第8週

2019/6/12

【Point 1】メンバの可視性に**public**を指定すると、クラスの外部から参照することができるようになる。(35行目) が、フィールドは基本的に**private**にする。(p.264)


```
1: class Car {  
2:     public int num;  
3:     private double gas;  
4:  
5:     public Car() {  
6:         num = 0; gas = 0.0;  
7:     }  
8:     private void setNum(int n) {  
9:         num = n;  
10:    }  
11:    public int getNum() {  
12:        return num;  
13:    }
```



【Point 2】**private**を指定すると、参照できるのはクラスの内部のみで、クラスの外部からは参照することができなくなる。(36行目) が、メソッドは基本的に**public**にする。(p.262)


【Point 3】フィールドに値を代入するメソッドは「セッタ(setter)」と呼ばれ、「set + フィールド名」と名前を付ける習慣がある。  
この例では、このセッタを使うことで、フィールドgasには指定した範囲内の値の代入が保証されることになる。

```
14:      public void setGas(double g) {  
15:          if(g > 0 && g < 1000 ) {  
16:              gas = g;  
17:          } else {  
18:              System.out.println("Out of range.");  
19:          }  
20:      }
```



【Point 4】フィールドの値を取得するメソッドは「**ゲッタ** (getter)」と呼ばれ、「**get + フィールド名**」と名前を付ける習慣がある。

```
21:    public double getGas() {  
22:        return gas;  
23:    }  
24:    public void show() {  
25:        System.out.println(" (num)" + num  
                               + " (gas)" + gas);  
26:    }  
27: }  
28:
```





```
29: class Pd8car1 {
30:     public static void main(String[] args) {
31:         Car c1 = new Car(); c1.show();
32:         c1.setGas(500); c1.show();
33:         c1.setGas(-100); c1.show();
34:
35:         c1.num = 1234;
36:         //c1.setNum(5678); //Error
37:         c1.show();
38:     }
39: }
```

【Point 5】フィールドと、それを使って処理するメソッドを1つのクラスにまとめることを「**カプセル化**」と呼び、フィールドをprivateとし、それにアクセスするメソッド(**アクセッサ** (accessor) = セッターとゲッターの総称)を用意することを「**情報隠蔽** (データ隠蔽)」と呼ぶ。(p.266)

【Point 6】public, privateを省略すると、「**package**」を指定したことになる。packageは**同じパッケージ内** (同じフォルダ内) のクラスからのみアクセス可能な可視性となる。(p.267)

# 【課題の準備】

演習室で作業する前に、以下のコマンドを  
入れるだけで準備が完了する

```
$ mygitclone4d 「自分のGitHubユーザ名」  
$ cd prog4d-(ユーザ名)  
$ ./myconf
```

※本体をシャットダウンするまでは、  
上記「mygitclone」と「myconf」の設定は有効です

# 【課題の準備】

以下の流れで、課題のプログラムを作るためのフォルダを準備しましょう。

1. 端末を起動して、以下のコマンドを実行して前期第8週のフォルダを作る  
\$ cd prog4d-(ユーザ名) (←既に移動しているなら不要)  
\$ mkdir week108  
\$ cd week108

※課題で作るファイル名は各自で決めて構いません。

# 【練習8-1】

サンプルプログラム「1\_08\_Car.java」を  
コンパイルして、実行結果を確認しましょう。



# 【課題8-1】

サンプルプログラムの36行目のコメントを外して  
コンパイルし、**どのようなエラーが出力されるのか**  
確認してください。

# 【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題8-1提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog4d-2019/prog4d-\(ユーザー名\)](https://github.com/nit-ibaraki-prog4d-2019/prog4d-(ユーザー名))

# 【課題8-2】

36行目のエラーの原因となっているメソッド  
setNumをpublicとなるようにプログラムを修正  
し、動作を確認してください。  
(その際、35行目は必要なくなる。)

# 【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m "課題8-2提出"
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog4d-2019/prog4d-\(ユーザー名\)](https://github.com/nit-ibaraki-prog4d-2019/prog4d-(ユーザー名))

# 【課題8-3】

メソッドsetNumを「車のナンバー（つまりフィールドnum）が1～9999の範囲である」という条件を満たすように改良して、動作を確認してください。

[mainを持ったクラス]

```
class Pd8car2 {  
    public static void main(String[] args) {  
        Car c1 = new Car();  
        c1.setNum(1234);  
        c1.show();  
        c1.setNum(77777);  
        c1.setNum(-1111);  
        c1.show();  
        c1.setNum(9999);  
        c1.show();  
    }  
}
```



# 【課題8-3】

## [ 実行結果 ]

```
(num)1234 (gas)0.0
```

```
Out of range.
```

```
Out of range.
```

```
(num)1234 (gas)0.0
```

```
(num)9999 (gas)0.0
```

( ← 77777を代入しようとした場合 )

( ← -1111を代入しようとした場合 )

( ← 9999を代入した結果 )

# 【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題8-3提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog4d-2019/prog4d-\(ユーザー名\)](https://github.com/nit-ibaraki-prog4d-2019/prog4d-(ユーザー名))