

Methods

Bad programming practice (1)

```
Cat kitty = new Cat();  
kitty.age = -2;
```

Cat
+age: int

Bad programming practice (2)

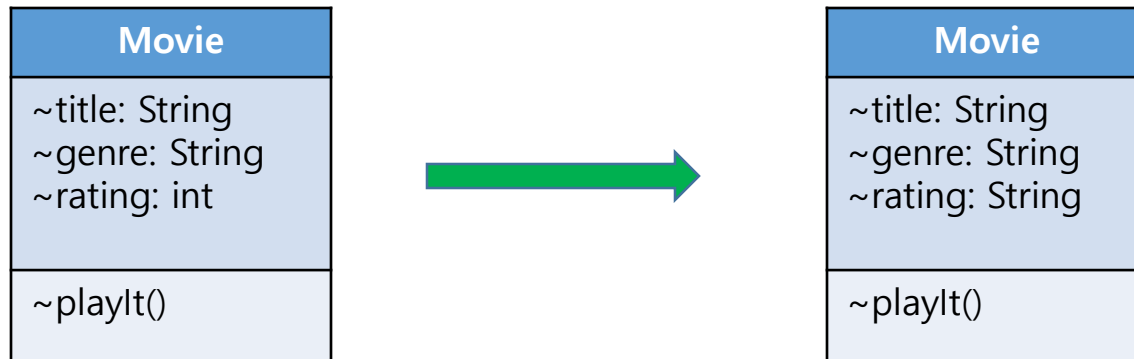
```
Movie myMovie = new Movie();  
  
myMovie.rating = -2;    // wrong!
```

Movie
~title: String ~genre: String ~rating: int
~playIt()

Bad programming practice (3)

```
Movie myMovie = new Movie();
```

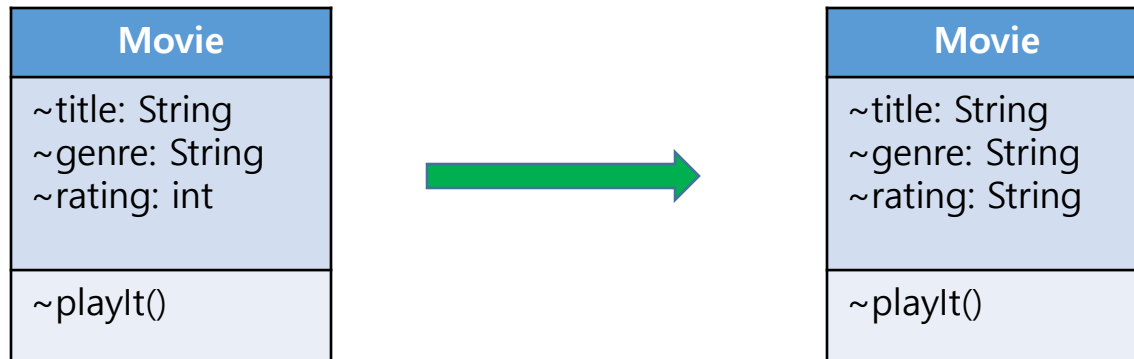
```
myMovie.rating = 4;           // used at 1000 places
```



Bad programming practice (3)

```
Movie myMovie = new Movie();
```

```
myMovie.rating = 4;           // used at 1000 places  
myMovie.rating = "****";      // change them all
```



How to Correct

- Movie class

```
class Movie {  
    String title;  
    String genre;  
    int rating;  
  
    void PlayIt() {  
        System.out.println("Playing the movie");  
    }  
}
```

Movie
~title: String ~genre: String ~rating: int
~playIt()

Encapsulation

- Two Different Meanings (from Wikipedia)
 1. A language construct that facilitates the bundling of data with the methods (or other functions) operating on that data.
 - class, structure with APIs(in C)
 2. A language mechanism for restricting access to some of the object's components
 - public, private, protected, package.
 - Purpose: Information hiding.

Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
<i>no modifier</i>	Y	Y	N	N
private	Y	N	N	N

How to Correct(2)

- Movie class

```
class Movie {  
    private String title;  
    private String genre;  
    private int rating;  
  
    void PlayIt() {  
        System.out.println("Playing the movie");  
    }  
}
```

Restrict visibility of all
instance variables to **private**.

Movie
-title: String -genre: String -rating: int
~playIt()


How to Correct (3)

- Restrict visibility of all instance variables to private.

```
class Movie {  
    private String title;  
    private String genre;  
    private int rating;  
  
    public void playIt() {  
        System.out.println("Playing " + title + " with rating " + rating);  
    }  
    public String getTitle() { return this.title; }  
    public String getGenre() { return this.genre; }  
    public int getRating() { return this.rating; }  
    public void setTitle(String title) {  
        this.title = title;  
    }  
    public void setGenre(String genre) {  
        this.genre = genre;  
    }  
    public void setRating(int rating) {  
        this.rating = rating;  
    }  
}
```

```
Movie myMovie = new Movie();  
myMovie.setRating(-1);  
myMovie.setTitle(null);
```

```
Movie myMovie = new Movie();  
myMovie.rating = -1;  
myMovie.title = null;
```



Movie

-title: String
-genre: String
-rating: int

+playIt()
+getTitle(): String
+getGenre(): String
+getRating(): int
+setTitle(newTitle: String)
+setGenre(newGenre: String)
+setRating(newRating: int)

How to Correct (4)

Movie
-title: String -genre: String -rating: int
+playIt() +getTitle(): String +getGenre(): String +getRating(): int +setTitle(newTitle: String): boolean +setGenre(newGenre: String): boolean +setRating(newRating: int): boolean

class name

member variables
(fields; instance variables)

methods

Accessors (getter)

Mutators (setter)

How to Correct (4)

Movie
-title: String -genre: String -rating: String
+playIt() +getTitle(): String +getGenre(): String +getRating(): String +setTitle(newTitle: String): boolean +setGenre(newGenre: String): boolean +setRating(newRating: int): boolean

class name

member variables

```
boolean setRating(int newRating) {  
    if (newRating < 1 || newRating > 6)  
        return false;  
    rating = newRating;  
    return true;  
}
```

int rating;

```
boolean setRating(int newRating) {  
    if (newRating < 1 || newRating > 6)  
        return false;  
    rating = "";  
    for (int i = 0; i < newRating; i++)  
        rating = rating+"*";  
    return true;  
}
```

String rating;

How to Correct (4)

Movie
-title: String -genre: String -rating: String
+playIt() +getTitle(): String +getGenre(): String +getRating(): String +setTitle(newTitle: String): boolean +setGenre(newGenre: String): boolean +setRating(newRating: int): boolean +setRating(newRating: String): boolean

class name

member variables
(fields; instance variables)

methods

Accessors (getter)

Mutators (setter)

Methods: example (1)

```
public class Movie {  
    // ...  
    public String getTitle() {  
        return this.title;  
    }  
  
    public boolean setTitle(String newTitle) {  
        if (newTitle == null) return false;  
        this.title = newTitle;  
        return true;  
    }  
  
    // ...  
}
```

Methods: example (2)

```
public class Movie {  
    // ...  
  
    public boolean setRating(int newRating) {  
        if (newRating < 1 || newRating > 6) {  
            return false;  
        }  
        this.rating = newRating;  
        return true;  
    }  
  
    // ...  
}
```

Methods: example (3)

```
Movie theBirds = new Movie();

theBirds.setTitle("The Birds");
theBirds.setGenre("Horror");
theBirds.setRating(3); // returns true
System.out.println(
    "Title: " + theBirds.getTitle()); // The Birds

theBirds.setTitle(null);
System.out.println(
    "Title: " + theBirds.getTitle()); // The Birds

theBirds.setRating(-1); // returns false
System.out.println(
    "Rating: " + theBirds.getRating()); // 3
```

Methods: example (4)

```
public class Cat {  
    private int age;  
  
    public int getAge() { return this.age; }  
    public boolean setAge(int inAge) {  
        if (inAge < 0) return false;  
        this.age = inAge;  
        return true;  
    }  
}
```

Cat
-age: int
+getAge(): int +setAge(inAge: int): boolean

Methods: example (4)

```
Cat kitty = new Cat();  
  
kitty.age = -2;    // invalid access.  
kitty.setAge(-2);
```

Cat
-age: int
+getAge(): int +setAge(inAge: int): boolean

Methods: example (5)

- Another design

```
public class Cat {  
    private int age;  
  
    public Cat() {  
        this.age = 0;  
    }  
    public int getAge() {  
        return this.age;  
    }  
    public void increaseAge() {  
        this.age++;  
    }  
}
```

Cat
-age: int
+getAge(): int +increaseAge()

Another example

Cat
-age: int -region: String
+Cat(at: String) +getAge(): int +setAge(inAge: int): boolean +increaseAge() +getRegion(): String

Another example

```
public class Cat {  
    private int age;  
    private String region;  
  
    public Cat() {  
        this.age = 0;  
        region = "Korea";  
    }  
  
    public Cat(String at) {  
        this.age = 0;  
        region = (at == null) ? "Korea" : at;  
    }  
    // continue...
```

Another example

```
// class Cat
public int getAge() {
    return (this.region.equals("Korea")
        ? (this.age + 1) : this.age);
}

public void increaseAge() {
    this.age++;
}

public String getRegion {
    return this.region;
}
}
```

References

- Kathy Sierra and Bert Bates, *Head First Java*, O'Reilly, 2005.

Q&A