Encapsulation

Encapsulation is an *object-oriented programming principle*.

Contents

Overview

Of the four *object-oriented programming principles*, **encapsulation** states that data (*variables*) should be bundled together with the code that operates on that data (*methods*), rather than allowing *direct* access.

Generally, this is done by *restricting access* to the *variables* within a class, and only allowing access through the *methods* of their current class.

To achieve encapsulation, then:

- make the class variables *private*
- provide *public* accessors to these variables: `get`ters and `set`ters

Encapsulation in action

Let's say we've got a class with directly-accessible variables in it:

```
public class ExtremelyImportantStuff {
    public int passportNumber = 1355417;
    public double bankBalance = -537.86;
    public String memorableWord = "pareidolia";
}
```

The issue with having a class like this is that you can change the value of each variable directly, which could cause problems later:

- we have less control over the variables and the methods which utilise them
- classes which do not need to use these variables will still be able to 'see' them
- every class has full access to these variables, including *read* and *write* functionality

Let's improve this:

```
public class ExtremelyImportantStuff {
    private int passportNumber = 1355417;
    private double bankBalance = -537.86;
    private String memorableWord = "pareidolia";

    public int getPassportNumber(){
        return passportNumber;
    }

    public void setPassportNumber(int newPassportNumber){
        this.passportNumber = newPassportNumber;
    }

    //etc.
}
```

The `get` method returns the value of the variable name.

The `set` method takes in a parameter and assigns it to the `name` variable - we use `this` to refer to the current object.

Read and Write-Only modules

We can make a module *read-only* by just using `get` methods:

```
public class Delegate {
    private String company = "QA";

    public String getCompany(){
        return company;
    }
}
```

We can make a module *write-only* by just using `set` methods:

```
public class Trainer {
    private String cohort;

    public void setCohort(String newCohort){
        this.cohort = newCohort;
    }
}
```

Tutorial

There is no tutorial for this module.

Exercises

Refactor your old stuff!

Go back through some of your other code and *encapsulate* your variables.

You may find that changing your code's functionality will be the hardest part.