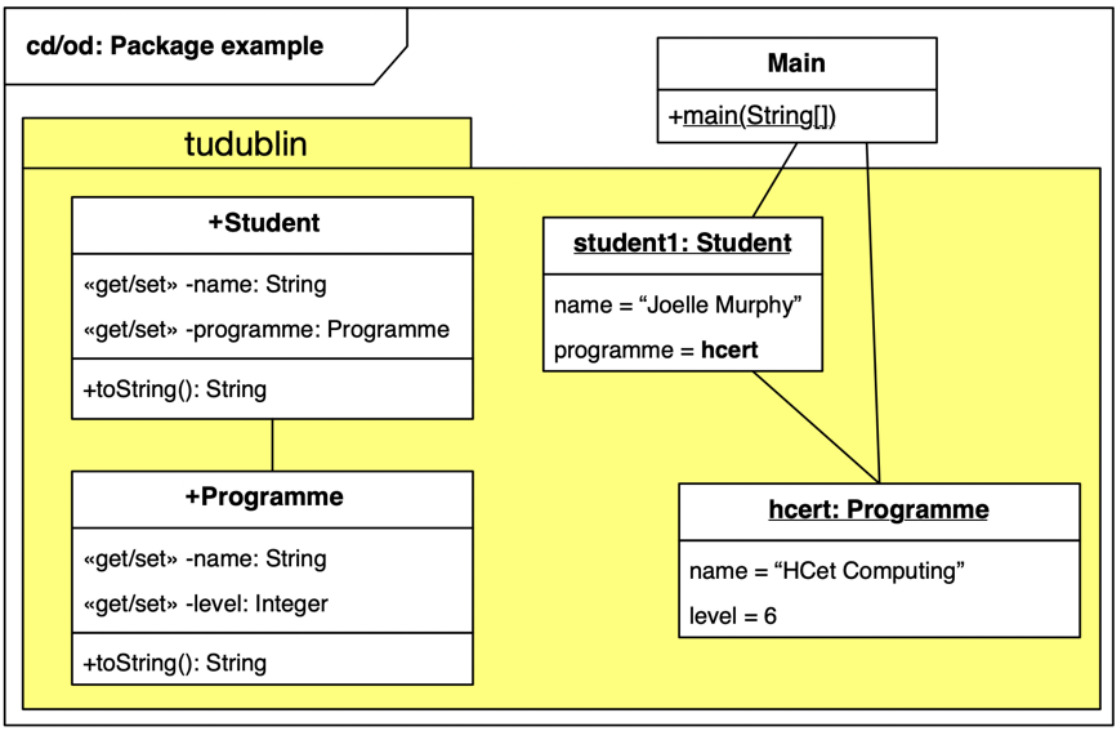
OO Programming

Lab 06

Object Oriented Programming
- Week 6 Java exercises

Packages

Exercise 1 - declaring a class in your package tudublin (3.20 from Book 2)



Class-Object diagram for package `tudublin`.

AIM:

- practice declaring classes in packages, and working with classes defined in other packages

ACTION:

- create a folder for package `tudublin`
- in your folder, create the following 2 classes for package `tudublin`
    - class `Programme` (File: `tudublin/Programme.java`)
        - public class `Programme` with private properties and public getters/setters:
            - this class represents University **programmes** of study (courses, like the Higher Certification in Computing and the BSc(Hons) degree in computing)
            - `name`: String
            - `level`: Integer
    - class `Student` (File: `tudublin/Student.java`)
        - public class `Student` with private properties and public getters/setters:
            - `name`: String
            - `programme`: Programme (a reference to a Programme object)
- class `Main` (File: `Main.java`)
    - method `main()`:
        - import all classes from package `tudublin`
        - create a `Programme` object `hcert` with name = `HCert Computing` and level = `6`
        - create a `Student` object `student1` with name = `Joelle Murphy` and programme linking to object `hcert`
        - print out each object via the default `toString()` which should confirm the package and class name of the object

OUTPUT:

```
$ javac tudubln/*.java
$ javac *.java
$ java Main
tudublin.Student@6d06d69c
tudublin.Programme@7852e922
```

ACTION - MORE:

- Now add custom `toString()` methods to each class, so the output describes each object as shown
    - the `toString()` method for `Student` can make use of the `toString()` method for `Programme` objects …
    - try to get the output to match the following:

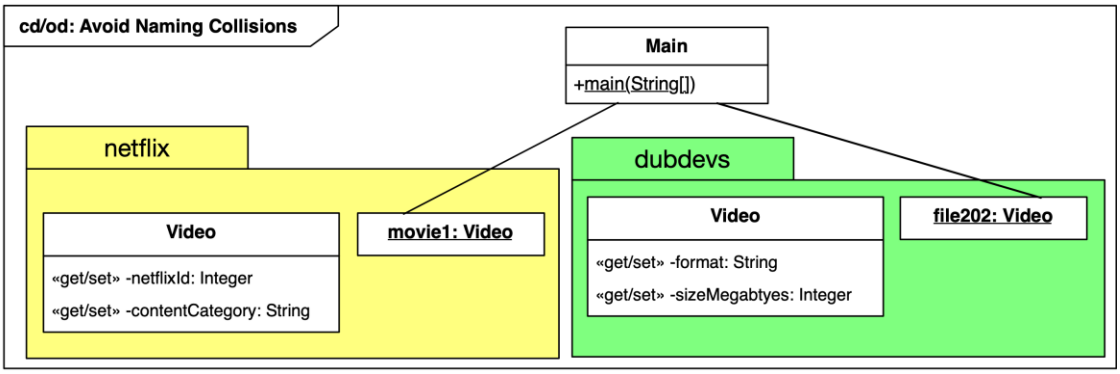
OUTPUT:

```
$ java Main
(Programme) name = HCert Computing / level = 6
(Student) name = Joelle Murphy /  programme = (Programme) name = HCert Com
puting (level 6)
```

HINT:

```
hcert = new Programme();
...
student1.setProgramme( hcert );
```

Exercise 2 - classes with same name in different packages (3.21 from Book 2)



Class-Object diagram for packages `netflix` and `dubdevs`.

AIM:

- use packages to avoid naming collisions where 2 classes have the same name

ACTION:

- imagine a Dublin software development company (with package name `dubdevs`) has been asked to work on some software for streaming video company **Netflix**, and both organisations have a class named `Video` …
- create a folder for package `netflix`
- in your folder `netflix`, create the following class for package `netflix`
    - class `Video` (File: `netflix/Video.java`)
        - class `Video` with private properties and public getters/setters:
            - `netflixId`: Integer
            - `contentCategory`: String
- create a folder for package `dubdevs`
- in your folder `dubdevs`, create the following class for package `dubdevs`
    - class `Video` (File: `dubdevs/Video.java`)
        - class `Video` with private properties and public getters/setters:
            - `format`: String
            - `sizeMegabtyes`: Integer
- class `Main` (File: `Main.java`)
    - method `main()`:

        - use a combination of `import` and fully qualified package-class names to create:
            - `Video` object `movie1` of the `netflix` class
            - `Video` object `file2020` of the `dubdevs` class
        - print out each object - using the default `toString()` method inherited from top-level Java class `Object`
- compile the classes in the package folder first, then compile and run your `Main` class

OUTPUT:

```
$ java Main
netflix.Video@6d06d69c
dubdevs.Video@7852e922
```

Exercise 3 - explore how cannot create object of non-public class in a package (3.22 from Book 2)

AIM:

- understand that from a class like `Main` that is outside the package, we can only create object-instances of the classes in the package that are `public`

ACTION:

- make a copy of your solution for the previous exercise
- remove the `public` keyword for the declaration of the Netflix `Video` class

OUTPUT:

You should get a compiler error, since we cannot now create the `movie1` instance-object of the `Video` class in package `netflix`, since that class is no longer `public` …