PGR112 – Session 7 (of 24)

Object-oriented programming

Marcus Alexander Dahl / marcusalexander.dahl@kristiania.no

Today's topics of focus

- Inheritance
- Aggregation (introduction)
- *super* keyword
- *final* keyword

+ an extra focus on coding today!

Inheritance

- Inheritance is specified using the extends keyword.
- A sub-class can inherit from a super-class (parent-, base class)
  - Re-using code!
- **`super`** is used to call the constructor of the parent class
  - **`this`** can be used to refer to a class' own constructor, for example when overloading constructors

Inheritance – *«is a»-relationship*

- When using inheritance, we should do so with caution.
  - Inheritance is used when the classes are in a **«is a»-relationship**
    - Circle is a Shape
    - Cat is an Animal
    - Banana is a Fruit

- Inheritance creates a strong connection between the two classes.

Aggregation (introduction)

- We can achieve re-usability of code by using aggregation.
  - In comparison to inheritance, the classes have a **«has a»-relationship**, examples:
    - Bottle has a Liquid (within it)
      - A soda can could have a liquid in it
      - A bucket could have a liquid in it
      - A bowl could have a liquid in it
    - Student has an Address
      - Campus could have an address
      - Home could have an address
  - The point is that the class on the right side of the relationship, can in most cases be used across code to represent the same concept across classes. For example:
    - Different locations could use the same Address class to represent a physical location
    - Different containers could use the Liquid class to represent different liquids stored within the containers

super keyword

- Allows us to call the constructor of our parent class
- Must be the first line of code within a class constructor that extends another class.
- Cannot use local instance methods to run code as part of passing arguments to the super constructor, as the instance doesn’t exist yet!
- **this** can be used within constructors to refer to the class's own constructor, often used when overloading the constructor, instead of having to type the name of the class. Not necessary!

`final` keyword

- This keyword is used often used to create:
  - Constant variables (read only)
  - Prevent method overriding
  - Prevent inheritance
  - Placed to the left of data types (when declaring variables or methods), or to the left of the *`class`* keyword when defining classes.

Let us explore some code that can show-case this;

Let us write some code!

- Tips and tricks
  - Hover your mouse cursor, the tooltip might display the keybinding!
  - Settings -> Keymap -> (Editor Actions || Main Menu > Edit > Find)
    - Explore the list, you may find many shortcuts here:
      - Add Selection for Next Occurrence
      - Duplicate Line or Selection
  - Alt + 1 can toggle your view of project files!
- Now, lets get started;