



# Object Oriented Software Development

Week 8



# UML

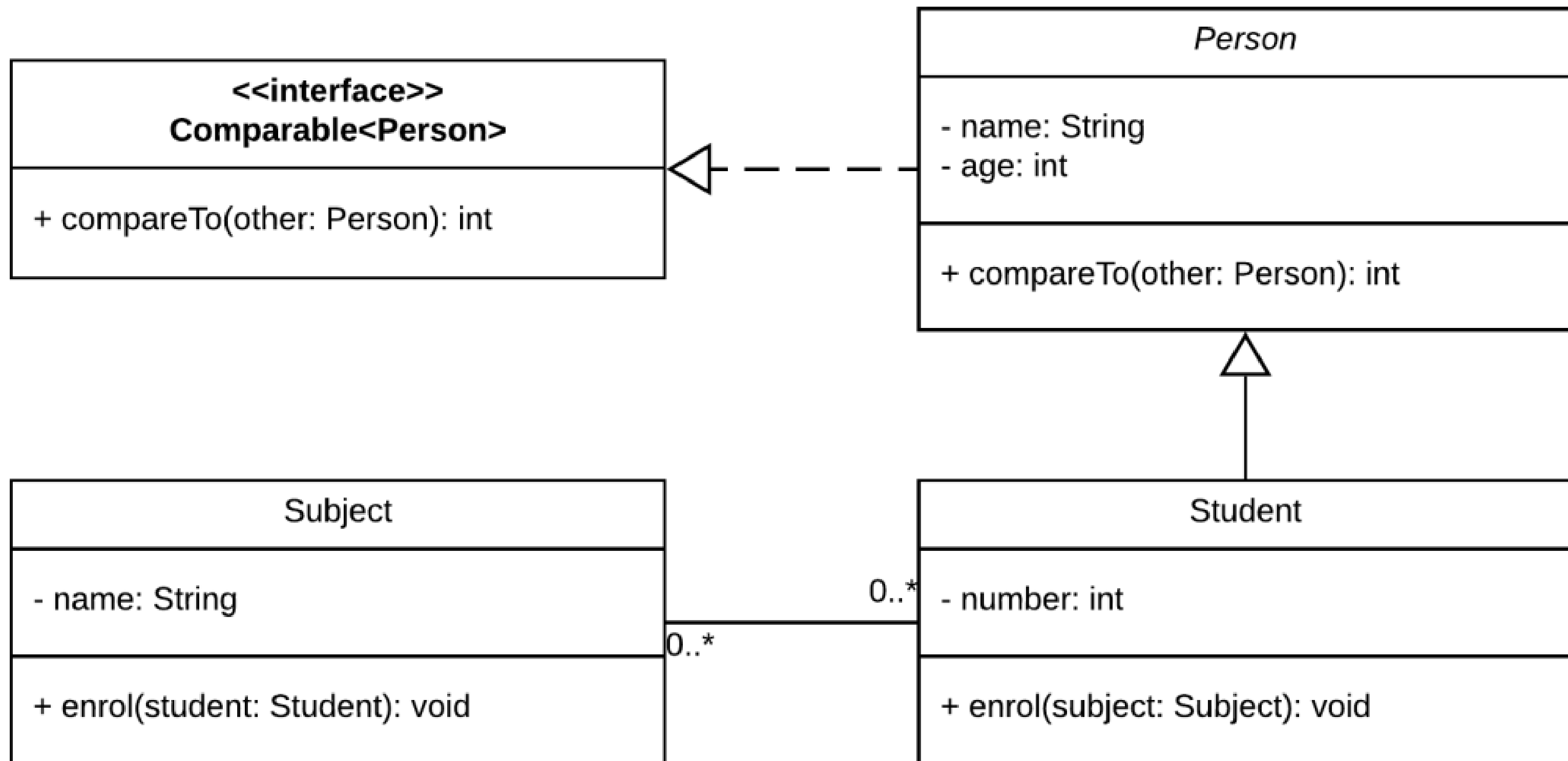
---

This week, we are learning to use UML class diagrams for designing and communicating our Java programs.

- A UML class diagram consists of classes (each of which have a name, zero or more attributes, and zero or more methods), as well as relationships between classes.
- Types of attributes and methods come **after** their name.
- An association from class A to class B means that A has an attribute of type B. You **must** use associations **instead of** attributes for classes that appear on your diagram. Classes that do not appear on your diagram (such as `String`) **do not need associations**.
- Associations come with a **multiplicity** representing how many instances of the type are stored (for example, in an array or `ArrayList`).
- Inheritance and interfaces are represented with an arrow **from** the subclass (or implementing class) **to** the parent class (or interface).
- Abstract classes and methods are represented using *italics*. Static attributes and methods are represented with an underline.
- Constructors, getters, and setters are not always shown on UML class diagrams.

There are many software options for creating UML class diagrams. We recommend <https://draw.io/>.

# UML Diagram



# Kahoot!

**Pairs or individually**

Share a single device

