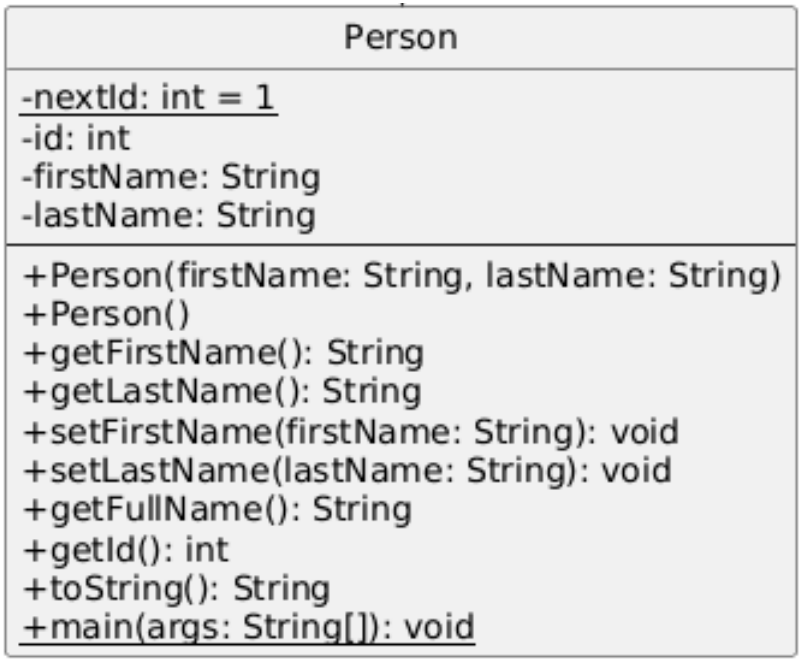
Inheritance, Abstract Classes, Interfaces~~, and Exceptions~~

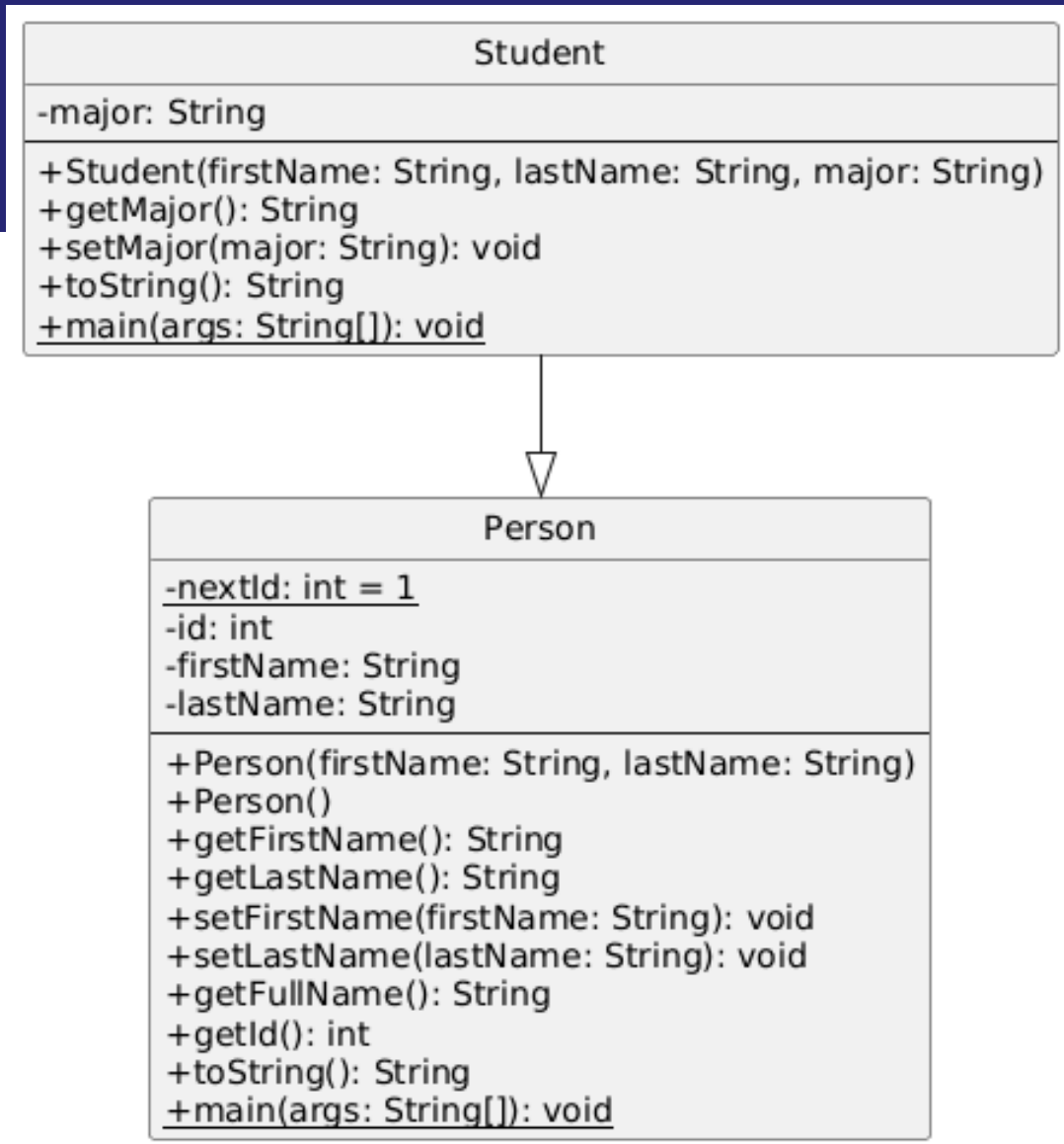
Inheritance

- A *subclass* (child/derived class) can inherit from a *superclass* (parent/base class)
- The child class “is-a” specialized type of the parent class
- The child class inherits all fields and methods from the parent
- The child class can add fields and add/overwrite methods



Inheritance

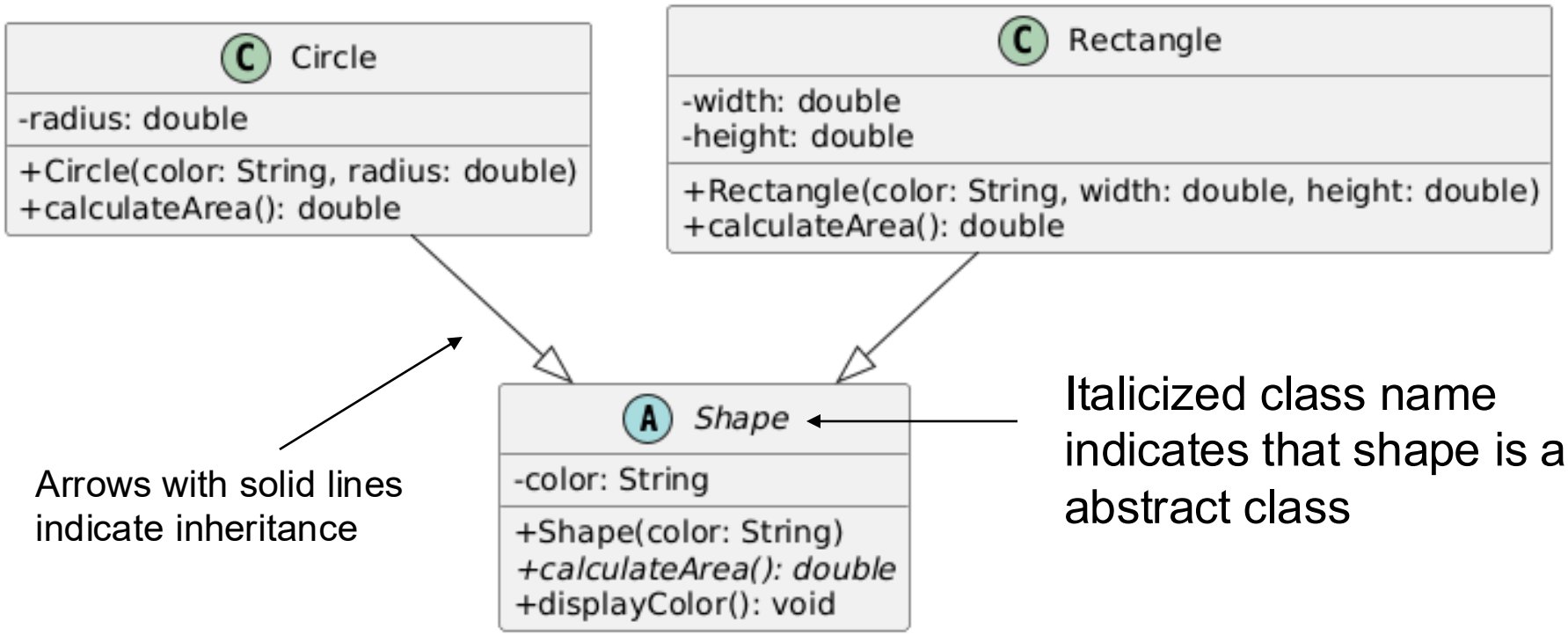
- A *subclass* (child/derived class) can inherit from a *superclass* (parent/base class)
- The child class "is-a" specialized type of the parent class
- The child class inherits all fields and methods from the parent
- The child class can add fields and add/overwrite methods



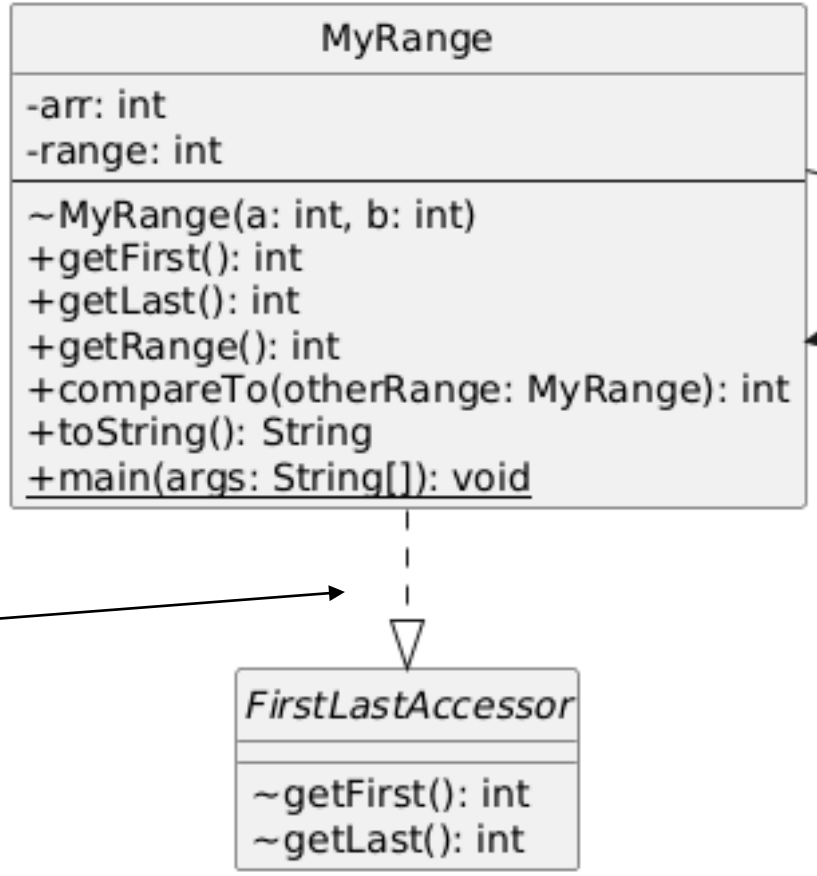
Abstract class basics

- An **abstract** class serves as a base template for a class and cannot be directly instantiated.
  - Abstract methods have no body and must be overwritten in a derived class
  - Concrete methods can be defined in the abstract class
- An **interface** serves as a blueprint specifying a collection of abstract methods that a class with the interface must implement.
- A single class can implement multiple interfaces, but a child class can inherit from only one parent class.

Abstract class example



Interface example



Dashed arrow indicates a self-dependency (a MyRange object temporarily interacts with another MyRange object)

Dashed arrow indicates MyRange implements an interface

Overview of UML diagrams:
https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-class-diagrams/

Generics

- Java Generics allow you to write classes, interfaces, and methods where the data type is specified as a parameter.

```
/**
 * Generic version of the Box class.
 * @param <T> the type of the value being boxed
 */
public class Box<T> {
    // T stands for "Type"
    private T t;

    public void set(T t) { this.t = t; }
    public T get() { return t; }
}
```

This type is optional (e.g., <> is allowed)

// instantiation

```
Box<Integer> integerBox = new Box<Integer>();
```

https://docs.oracle.com/javase/tutorial/java/generics/types.html

ArrayList

- A Java ArrayList behaves like a resizable, dynamic array that can store any type of object.
- Example usage: https://www.w3schools.com/java/java_arraylist.asp
- Documentation: https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html