CSC207 - Object Oriented Design

Ilir Dema

Summer 2016

UML

- Unified Modeling Language (UML) allows us to express the design of a program before writing any code.
- It is language-independent.
- An extremely expressive language.
- We'll use only a small part of th elanguage, Class Diagrams to represent basic OO design.

Example: Class Player

Each class is represented by a box divided in three sections:

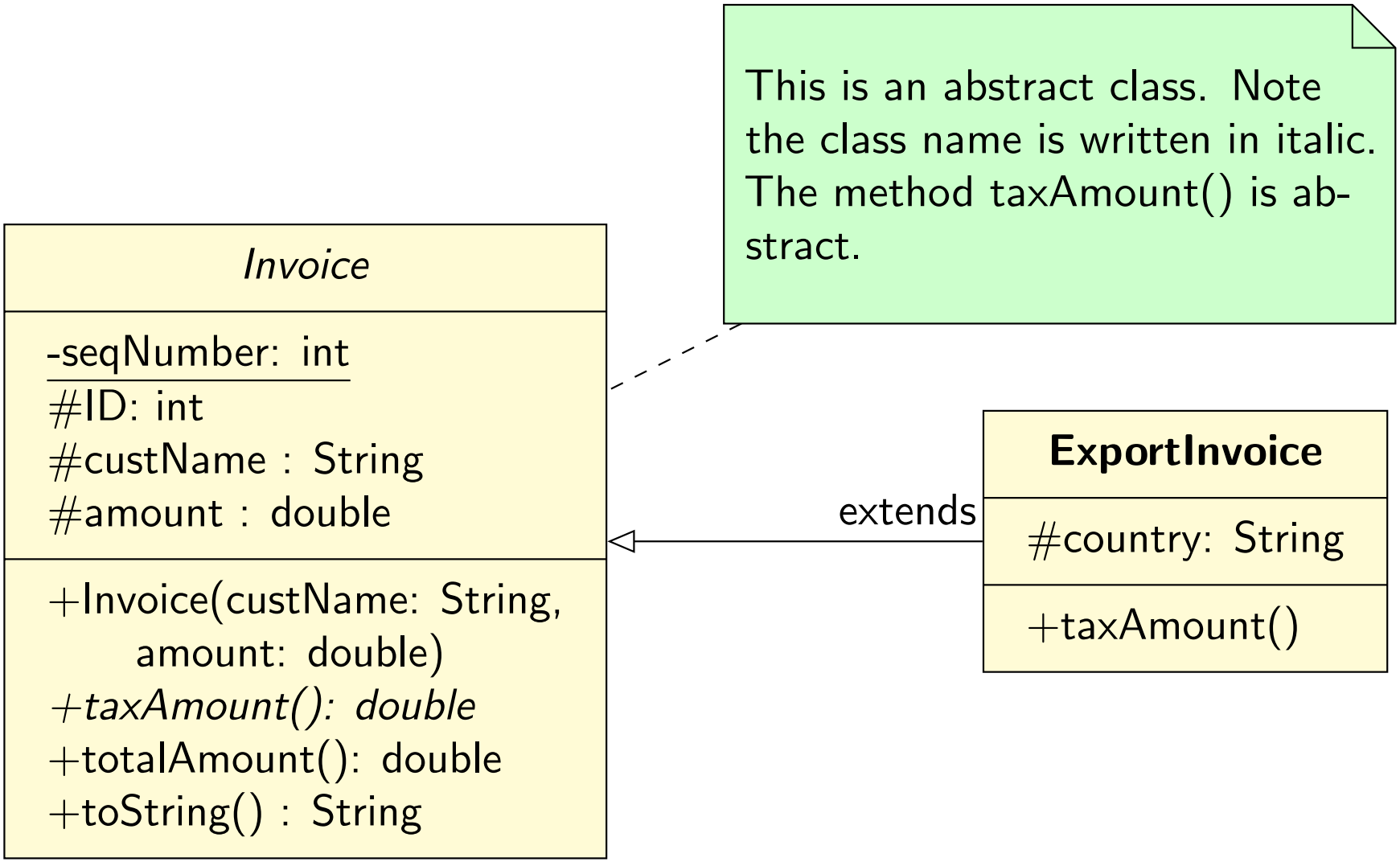
- Class name
- Data members
- Methods

Player
-name : String -manaPoints : int -healthPoints: int
+Player(name: String, manaPoints: int, healthPoints: int) +getName() :String +fight(other: Player) : void +toString() : String

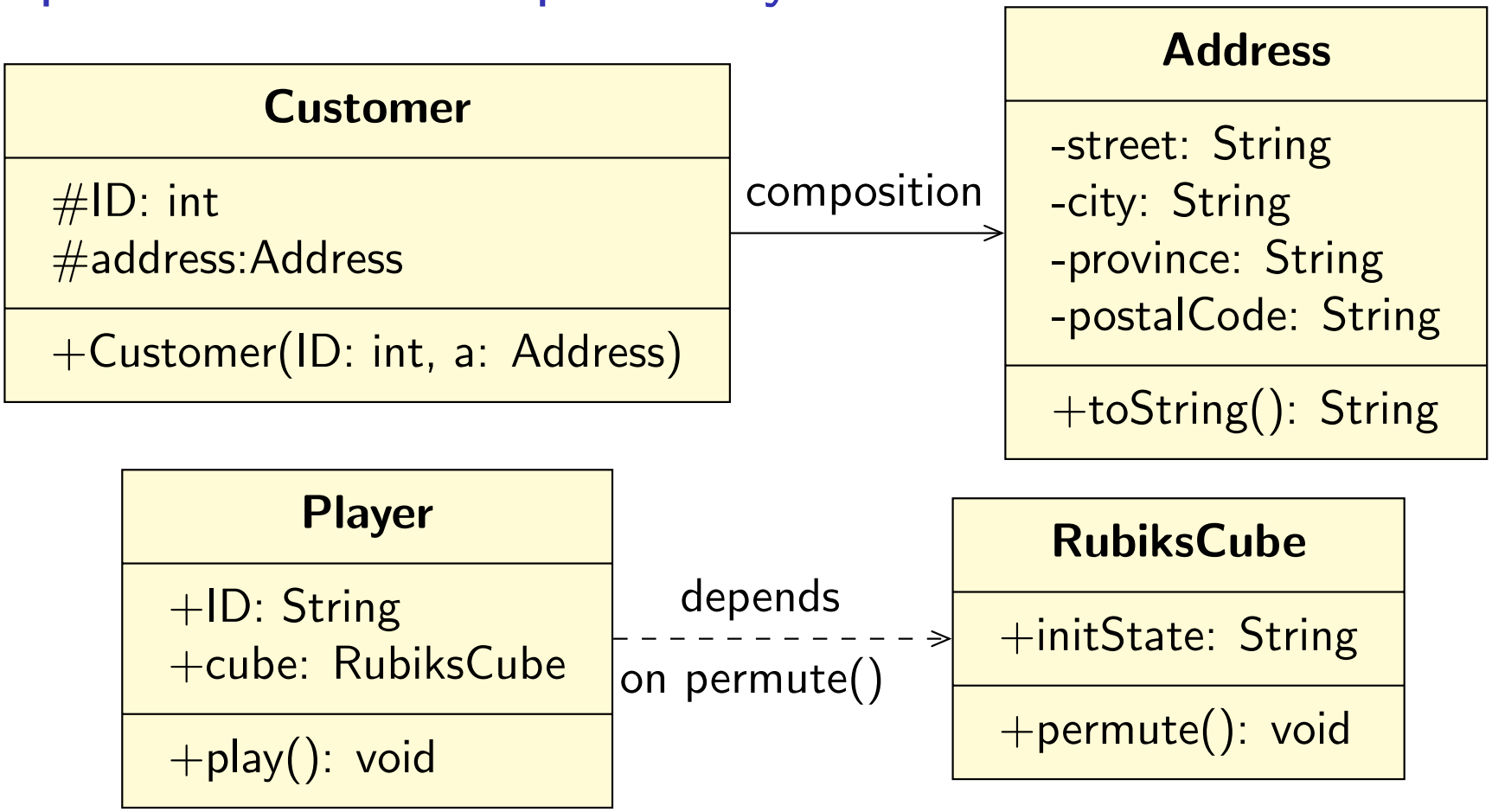
Notation

- Data Members:
  - `name: type`
- Methods:
  - `methodName(param1: type1, param2: type2, ...): returnType`
- Visibility:
  - $-$ private
  - $+$ public
  - $\#$ protected
  - $\sim$ package
- Static: underline
- Abstract method: *italic*
- Abstract class: *italic* or <<abstract>>
- Interface: <<Interface>>

Abstract Classes and Inheritance



Composition versus dependency




```
Class Player {
// ....
play() {
    cube.permute(); // Calling permute }
}
```

Interfaces and generics