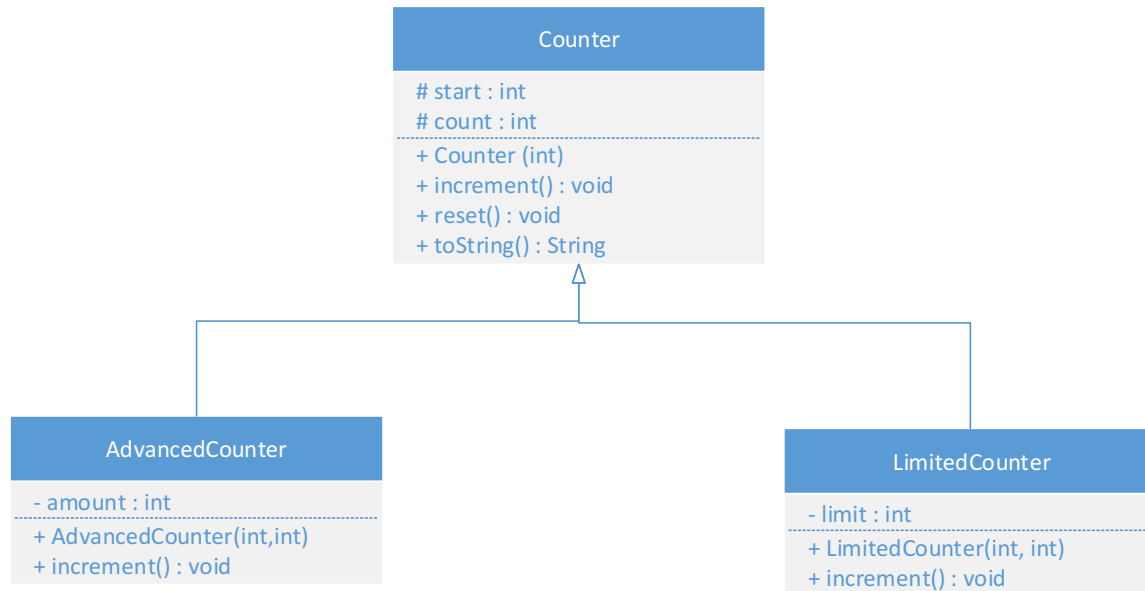


King Saud University  
College of Computer and Information Sciences  
Department of Computer Science  
CSC113 – Computer Programming II – Tutorial: Inheritance – Spring 2019



### Counter class:

#### Attributes:

- `start`: starting value of the counter
- `count`: the current value of the counter

#### Methods:

- `Counter(value: int)`: constructor
- `increment()`: this method increments the count by one
- `reset()`: this method resets the count to its starting value
- `toString()`: this method returns the object info in the following format `'( count )'`

### AdvancedCounter class:

#### Attributes:

- `amount`: the amount to increment the count with

#### Methods:

- `AdvancedCounter(value: int, amount: int)`: constructor
- `increment()`: this method increments the *count* by *amount*

### LimitedCounter class:

#### Attributes:

- `limit`: the upper limit that the counter stops at

#### Methods:

- `LimitedCounter(value: int, limit: int)`: constructor
- `increment()`: this method increments the count by one only if the limit hasn't been reached. It prints a message saying the limit has been reached otherwise.

King Saud University  
College of Computer and Information Sciences  
Department of Computer Science  
CSC113 – Computer Programming II – Tutorial: Inheritance – Spring 2019

Exercise 1: Translate into Java-code the class Counter, AdvancedCounter, and LimitedCounter.

Exercise 2: Write a test class with a main method and do the following:

- create three objects of type Counter, AdvancedCounter and LimitedCounter
- increment the three of them once
- print the value of the counter for each
- increment the LimitedCounter object to its limit
- reset it and print its value