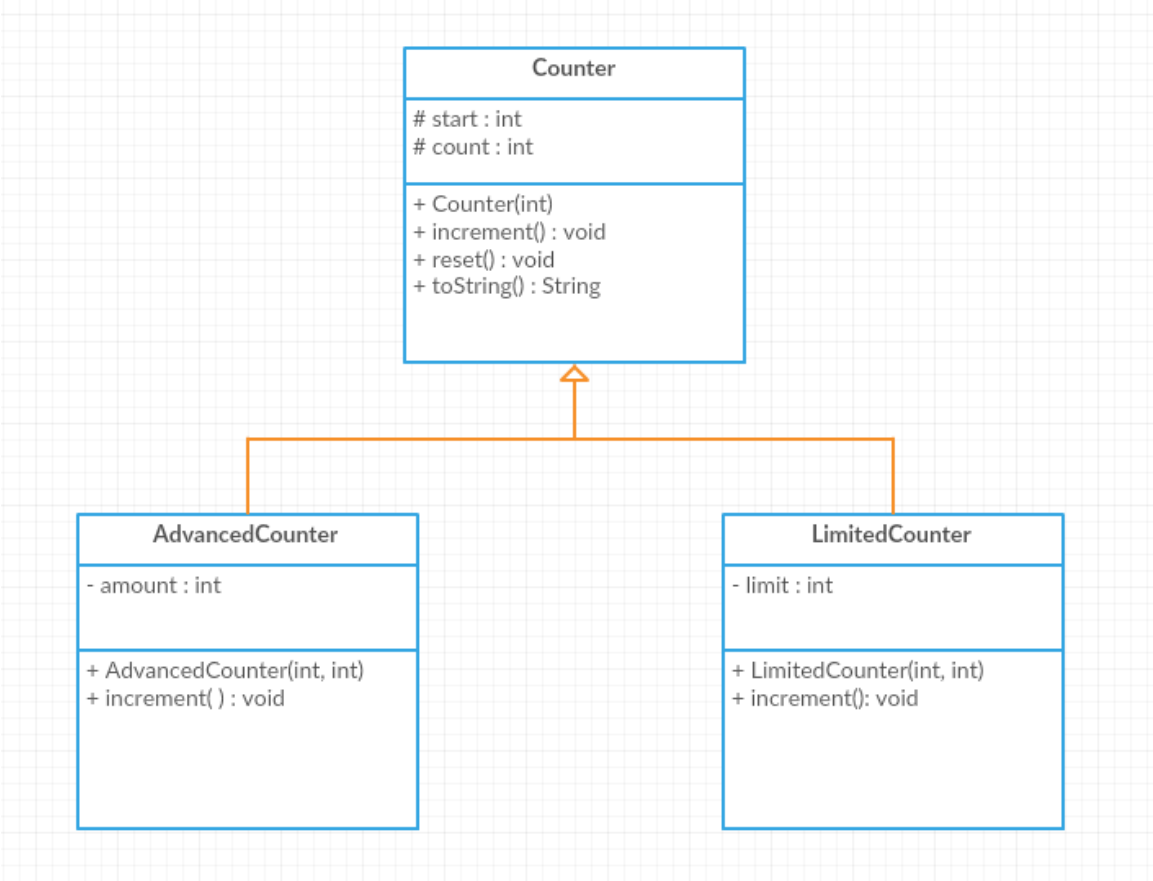
King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Tutorial No 5 – Spring 2017



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*

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Tutorial No 5 – Spring 2017

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.

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

Solution

```
public class Counter {
	protected int start;
	protected int count;

	public Counter(int value) {
		this.start = this.count = value;
	}

	public void increment() {
		count++;
	}

	public void reset() {
		count = start;
	}

	public String toString() {
		return "(" + count + ")";
	}
}
```

```
public class AdvancedCounter extends Counter {
	private int amount;

	public AdvancedCounter(int value, int amount) {
		super(value);
		this.amount = amount;
	}

	public void increment() {
		count += amount;
	}
}
```

```
public class LimitedCounter extends Counter {
	private int limit;

	public LimitedCounter(int value, int limit) {
		super(value);
		this.limit = limit;
	}

	public void increment() {
		if (count >= limit) {
			System.out.println("Reached limit, cannot increment
anymore.");
		}
		else {
			count++;
		}
	}
}
```

```
public class TestCounter {
	public static void main(String[] args) {
		Counter basic = new Counter(0);
		Counter limited = new LimitedCounter(0, 3);
		Counter advanced = new AdvancedCounter(0, 2);
		basic.increment();
		limited.increment();
		limited.increment();
		limited.increment();
		limited.increment();
		advanced.increment();
		System.out.println(basic.toString());
		System.out.println(limited.toString());
		System.out.println(advanced.toString());
		limited.reset();
		System.out.println(limited.toString());
	}
}
```