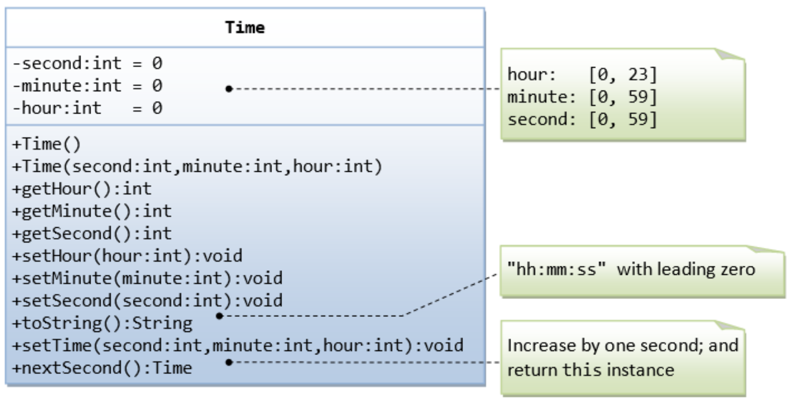
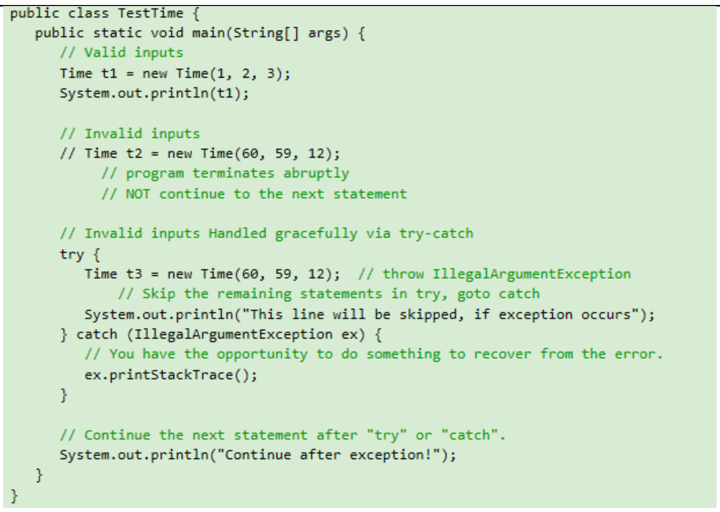
Modul 226

Handlungsziel	A	B	C

TITEL	**Zeit Applikation (Klasse)**
BESCHREIBUNG	 Time -second:int = 0 -minute:int = 0 -hour:int = 0 +Time() +Time(second:int,minute:int,hour:int) +getHour():int +getMinute():int +getSecond():int +setHour(hour:int):void +setMinute(minute:int):void +setSecond(second:int):void +toString():String +setTime(second:int,minute:int,hour:int):void +nextSecond():Time hour: [0, 23] minute: [0, 59] second: [0, 59] "hh:mm:ss" with leading zero Increase by one second; and return this instance
IDEE, BEISPIEL	Erstellen Sie eine Klasse Time und erstellen Sie dazu eine Test-klasse in der Sie die einzelnen Methoden testen können.
SCHWIERIGKEIT	A class called `Time`, which models a time instance with hour, minute and second, is designed as shown in the class diagram. It contains the following members: • 3 `private` Instanz `hour`, `minute`, and `second`. • Constructors, getters and setters. • A method `setTime()` to set `hour`, `minute` and `second`. • A `toString()` that returns "`hh:mm:ss`" with leading zero if applicable. • A method `nextSecond()` that advances `this` instance by one second. It returns `this` instance to support chaining (cascading) operations, e.g., `t1.nextSecond().nextSecond()`. Take note that the `nextSecond()` of `23:59:59` is `00:00:00`. Write the `Time` class and a test driver to test all the public methods.

SKALIERBARKEIT	Input Überprüfung mit *Exception Handling*
	`// Throw an exception if input is invalid`
	`public void setHour(int hour) {`
	`   if (hour >= 0 && hour <= 23) {`
	`      this.hour = hour;`
	`   } else {`
	`      throw new IllegalArgumentException("Invalid hour!");`
	`   }`
	`}`



```java
public class TestTime {
   public static void main(String[] args) {
      // Valid inputs
      Time t1 = new Time(1, 2, 3);
      System.out.println(t1);

      // Invalid inputs
      // Time t2 = new Time(60, 59, 12);
         // program terminates abruptly
         // NOT continue to the next statement

      // Invalid inputs Handled gracefully via try-catch
      try {
         Time t3 = new Time(60, 59, 12);  // throw IllegalArgumentException
             // Skip the remaining statements in try, goto catch
         System.out.println("This line will be skipped, if exception occurs");
      } catch (IllegalArgumentException ex) {
         // You have the opportunity to do something to recover from the error.
         ex.printStackTrace();
      }

      // Continue the next statement after "try" or "catch".
      System.out.println("Continue after exception!");
   }
}
```

VORGEHEN	1.Beschreibung, 2. Pflichtenheft, 3. AD/ZD, 4.Implementierung