

Task 1

GTv1 screenshot

The screenshot displays the IntelliJ IDEA IDE interface. The main editor shows the `GTv1` class in the `ee.ut.cs.swt.nextdate` package. The class contains three test methods: `testCreatesNextDate1`, `testCreatesNextDate0`, and `testCreatesNextDate3`. The `testCreatesNextDate1` method is currently selected and highlighted.

The Project view on the left shows the project structure, including the `src` directory, `main` and `test` sources, and the `target` directory. The `GTv1` class is highlighted in the `test` source.

The Coverage view on the right shows the coverage for the `GTv1` class. The table below summarizes the coverage data:

Element	Class, %	Method, %	Line, %
ee.ut.cs.swt.nextdate	100% (1/1)	100% (6/6)	87% (36/41)
NextDate	100% (1/1)	100% (6/6)	87% (36/41)

The bottom status bar shows the test results for the `GTv1` class. The tests passed, and the coverage report is displayed in the bottom right pane.

Tests passed: 21 of 21 tests - 29 ms

IntelliJ IDEA coverage runner

Line coverage ...

include patterns:

ee\ut\cs\swt\nextdate\..*

exclude patterns:

exclude annotations patterns:

.*Generated.*

Class transformation time: 0.0164107s for 511 classes or 3.211487279843444E-5s per class

MTv1 screenshot

The screenshot displays the IntelliJ IDEA IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, and Help. The toolbar shows icons for running tests and viewing coverage.

The **Project** view on the left shows the project structure:

- NDv1 [nextdate] C:\Users\kasnikov\Desktop\All about coding\testing\hw5\NDv1
 - .idea
 - src
 - main
 - java
 - ee.ut.cs.swt.nextdate 100% classes, 65% lines covered
 - NextDate 100% methods, 65% lines covered
 - test
 - java
 - ee.ut.cs.swt.nextdate
 - GTv1
 - MTv1
 - target
 - pom.xml
 - External Libraries
 - Scratches and Consoles

The **Code** view in the center shows the `NextDate.java` file with the following code:

```
1 package ee.ut.cs.swt.nextdate;
2
3 import ...
4
5
6 public class GTv1 {
7
8     @Test(timeout = 4000)
9     public void testCreatesNextDate1() throws Throwable {
10         NextDate nextDate0 = new NextDate(m: 10, d: 10, y: 10);
11         String string0 = nextDate0.run(month: 10, day: 31, year: 1970);
12         assertEquals(expected: "11/1/1970", string0);
13     }
14
15     @Test(timeout = 4000)
16     public void testCreatesNextDate0() throws Throwable {
17         NextDate nextDate0 = new NextDate(m: 12, d: 12, y: 12);
18         String string0 = nextDate0.run(month: 12, day: 1839, year: 1970);
19         assertEquals(expected: "invalid Input Date", string0);
20     }
21
22     @Test(timeout = 4000)
23     public void testCreatesNextDate3() throws Throwable {
24         NextDate nextDate0 = new NextDate(m: 9, d: 9, y: 9);
25         String string0 = nextDate0.run(month: 9, day: 31, year: 1970);
26         assertEquals(expected: "invalid Input Date", string0);
27     }
28 }
```

The **Coverage** view on the right shows the following data:

Element	Class, %	Method, %	Line, %
ee.ut.cs.swt.nextdate	100% (1/1)	100% (6/6)	65% (27/41)
NextDate	100% (1/1)	100% (6/6)	65% (27/41)

The **Run** view at the bottom shows the test results for `MTv1`:

- MTv1 (ee.ut.cs.swt.nextdate) 21 ms
- testDecemberToJanuary 20 ms
- testLeapYear 1 ms
- testJuneToJuly 0 ms
- testMarchToApril 0 ms

The console output shows the following error:

```
org.junit.ComparisonFailure:
Expected: "invalid Input Date"
Actual: "11/1/1970"
```

Task 2, screenshots

GTv1

The screenshot displays the IntelliJ IDEA IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, and Help. The status bar at the bottom shows NDV1, src > test > java > ee > ut > cs > swt > nextdate > GTv1, and testCreatesNextDate7.

The Project view on the left shows the project structure with the following folders and files:

- ee.ut.cs.swt.nextdate (100% classes, 81% lines covered)
 - NextDate (100% methods, 81% lines covered)
 - resources
 - test
 - java
 - ee.ut.cs.swt.nextdate
 - GTv1
 - GTv12
 - MTv1
 - target

The Structure view on the left shows the class structure for GTv1:

- testCreatesNextDate1(): void
- testCreatesNextDate0(): void
- testCreatesNextDate3(): void
- testCreatesNextDate2(): void
- testCreatesNextDate5(): void
- testCreatesNextDate4(): void
- testRunReturningNonEmptyString(): void
- testRunWithNegative(): void
- testRunWithNegativeAndNegative(): void

The main editor shows the code for GTv1, including the testCreatesNextDate7() method:

```
String string0 = nextDate0.run( month: 1, day: 1, year: 1);
assertEquals( expected: "invalid Input Date", string0);

@Test(timeout = 4000)
public void testCreatesNextDate7() throws Throwable {
    NextDate nextDate0 = new NextDate( m: 7, d: 7, y: 7);
    String string0 = nextDate0.run( month: 7, day: 7, year: 7);
    assertEquals( expected: "7/7/1938", string0);
}

@Test(timeout = 4000)
public void testCreatesNextDate6() throws Throwable {
    NextDate nextDate0 = new NextDate( m: 12, d: 12, y: 12);
    String string0 = nextDate0.run( month: 12, day: 12, year: 12);
    assertEquals( expected: "12/13/1839", string0);
}

@Test(timeout = 4000)
public void testCreatesNextDate9() throws Throwable {
    NextDate nextDate0 = new NextDate( m: 2, d: 2, y: 2);
    String string0 = nextDate0.run( month: 2, day: 28, year: 2);
    assertEquals( expected: "3/1/1879", string0);
}
```

The Coverage view on the right shows the coverage for GTv1:

Element	Class, %	Method, %	Line, %
ee.ut.cs.swt.nextdate	100% (1/1)	100% (7/7)	81% (40/49)
NextDate	100% (1/1)	100% (7/7)	81% (40/49)

The Run view at the bottom shows the test results for GTv1:

- testRunWithNegative: 8 ms
- testRunWithNegativeAndNegative: 1 ms
- testRunReturningNonEmptyString: 1 ms
- testRunWithPositive0: 1 ms
- testRunWithPositive1: 0 ms
- testRunWithPositive2: 0 ms
- testCreatesNextDate0: 0 ms
- testCreatesNextDate1: 8 ms
- testCreatesNextDate2: 0 ms
- testCreatesNextDate3: 0 ms

The test results summary shows: Tests passed: 21 of 21 tests - 23 ms. The output window shows the command: C:\Users\kasnikov\.jdk\openjdk-21.0.2\bin\java.exe ... and the coverage report details.

GTv12

The screenshot displays the IntelliJ IDEA IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, and Help. The top toolbar shows the NDV1 version control system and various development icons.

The main editor area is split into three panes:

- Project View:** Shows the project structure with folders for ee.ut.cs.swt.nextdate, resources, test, and target. The test folder contains subfolders for java and target. The java folder contains subfolders for ee.ut.cs.swt.nextdate, GTv1, GTv12, and MTv1.
- Structure View:** Shows the class structure for GTv12, listing methods like testCreatesNextDate1(), testCreatesNextDate0(), testCreatesNextDate3(), testCreatesNextDate2(), testCreatesNextDate5(), testCreatesNextDate4(), testRunWithNegative(), testCreatesNextDate14(), and testCreatesNextDate13().
- Code Editor:** Displays the source code for GTv12.java. The code defines a package ee.ut.cs.swt.nextdate, imports java.util.Date, and defines a public class GTv12. It includes three test methods: testCreatesNextDate1(), testCreatesNextDate0(), and testCreatesNextDate3().

The right sidebar shows the Coverage view for GTv12, displaying a table of coverage data:

Element	Class, %	Method, %	Line, %
ee.ut.cs.swt.nextdate	100% (1/1)	100% (7/7)	97% (48/49)
NextDate	100% (1/1)	100% (7/7)	97% (48/49)

The bottom pane shows the Coverage view for GTv12, displaying a table of test results:

Test Name	Duration
testRunWithNegative	8 ms
testRunWithNegativeAndNegative	1 ms
testRunReturningNonEmptyString	9 ms
testRunWithPositive0	1 ms
testRunWithPositive1	0 ms
testRunWithPositive2	1 ms
testCreatesNextDate0	1 ms
testCreatesNextDate1	0 ms
testCreatesNextDate2	0 ms
testCreatesNextDate3	1 ms

The bottom status bar shows the current file path: NDV1 > src > test > java > ee > ut > cs > swt > nextdate > GTv12 > testCreatesNextDate1. The bottom right corner shows the encoding: 13:6 CRLF UTF-8 Tab*.

MTv1

FileEditViewNavigateCodeRefactorBuildRunToolsVCSWindowHelp

NDv1Version control

Project

ee.ut.cs.swt.nextdate 100% classes, 65% lines covered

NextDate 100% methods, 65% lines covered

resources

test

java

ee.ut.cs.swt.nextdate

GTv1

GTv12

MTv1

target

Structure

MTv1

tearDownAfterClass(): void

testJuneToJuly(): void

testMarchToApril(): void

testDecemberToJanuary(): void

testLeapYear(): void

trialDate: NextDate = new NextDate(...)

NextDate.java

GTv12.java

GTv1.java

MTv1.java

Coverage MTv1

13

14

15

16

17

18

19

20

21

24

25

26

29

30

31

32

33

34

37

38

//@Test

/*

* Test method for 'ee.ut.cs.swt.nextdate.NextDate.run(int

*/

@Test

public final void testJuneToJuly() { assertEquals(expected:

@Test

public final void testMarchToApril() { assertEquals(expecte

@Test

public final void testDecemberToJanuary() { assertEquals(e

@Test

public final void testLeapYear() { assertEquals(expected: "

}

Element

Class, %

Method, %

Line, % ^

ee.ut.cs.swt.nextdate

NextDate

100% (1/1)

100% (7/7)

65% (32/49)

100% (1/1)

100% (7/7)

65% (32/49)

Cover

MTv1

MTv1 (ee.ut.cs.swt.nextdate) 12ms

testDecemberToJanuary 2ms

testLeapYear 10ms

testJuneToJuly 0ms

testMarchToApril 0ms

Tests passed: 4 of 4 tests - 12ms

C:\Users\kasnikov\.jdk\openjdk-21.0.2\bin\java.exe ...

---- IntelliJ IDEA coverage runner ----

Line coverage ...

include patterns:

ee\ut\cs\swt\nextdate\..*

exclude patterns:

exclude annotations patterns:

.*Generated.*

Class transformation time: 0.0132703s for 499 classes or 2.6593787575150303E-5s per class

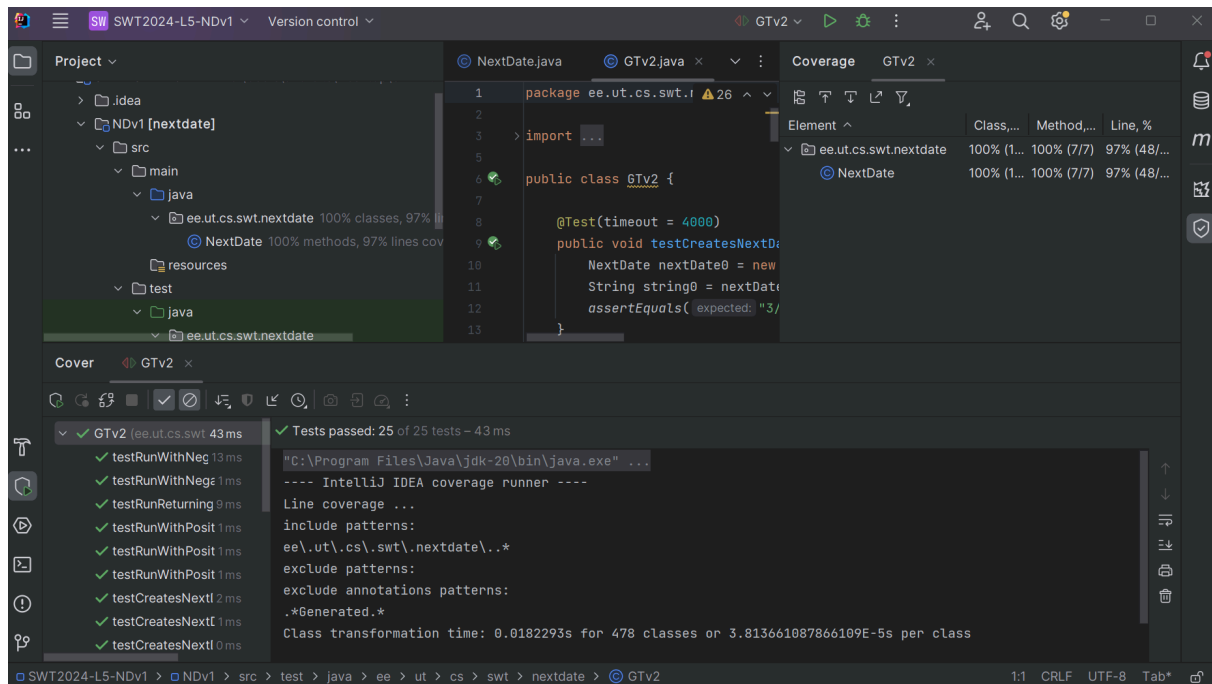
Done:ace finished with exit code 0

NDv1 > src > test > java > ee > ut > cs > swt > nextdate > MTv1

38:1 CRLF UTF-8 Tab*

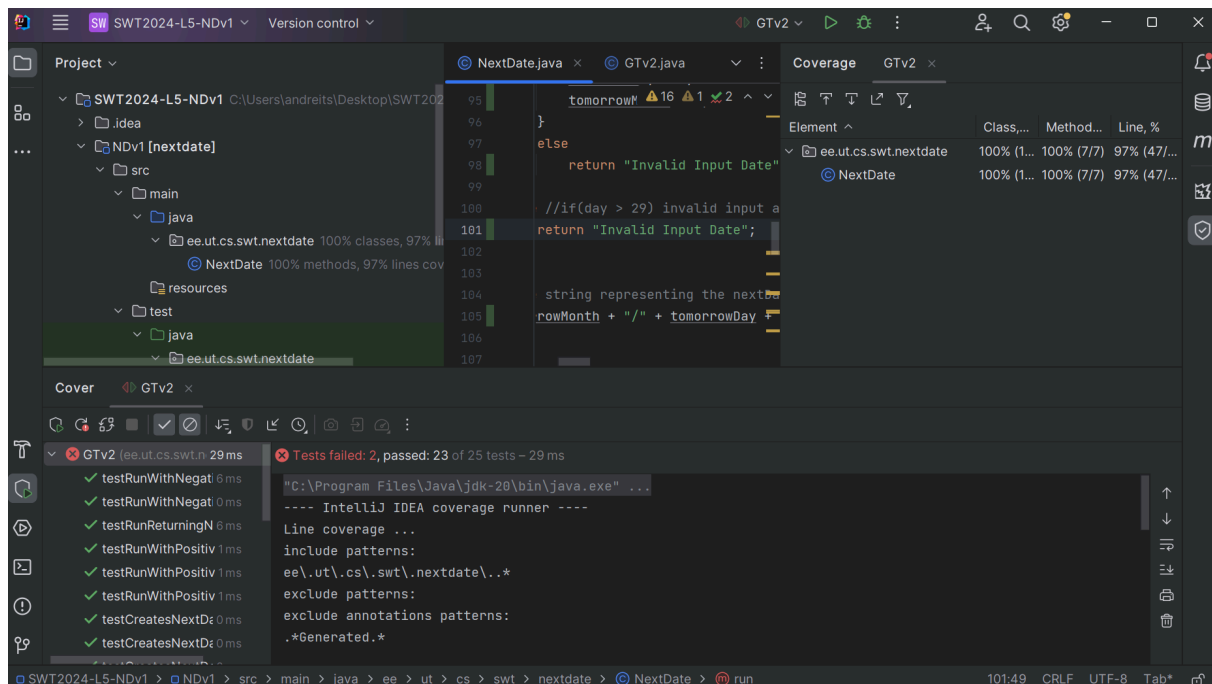
Task 3

GTv2



Task 4

GTv2



MTv1

The screenshot displays the IntelliJ IDEA IDE with the following components:

- Project View:** Shows the project structure with folders like `resources`, `test`, and `java`. The `ee.ut.cs.swt.nextdate` package is expanded, showing classes `GTv1`, `GTv2`, `GTv12`, and `MTv1`.
- Code Editor:** Displays the `NextDate.java` file. The code includes a `tomorrowM` variable, an `else` block returning `"Invalid Input Date"`, and a comment `/*if(day > 29) invalid input a`. Line numbers 95 to 107 are visible.
- Coverage Table:** Shows coverage data for `ee.ut.cs.swt.nextdate` and `NextDate`.

Element	Class, %	Method, %	Line, %
<code>ee.ut.cs.swt.nextdate</code>	100% (1...)	100% (7/7)	70% (34/...
<code>NextDate</code>	100% (1...)	100% (7/7)	70% (34/...
- Test Results:** Shows the execution of `MTv1` with 1 test failed and 3 passed. The failed test is `testLeapYear`. The output pane shows the command `"C:\Program Files\Java\jdk-20\bin\java.exe" ...` and the IntelliJ IDEA coverage runner output.