**Assignments on Junit**

1) Write a class called MinMaxFinder. Define a method in it called findMinMax() which

accepts an int array and returns new array of size 2, wherein the 0th index will have the

min value of the array and 1st index will have max value of the array. Perform Junit testing

of the method findMinMax with as many test cases you can think of (min 3 test cases)

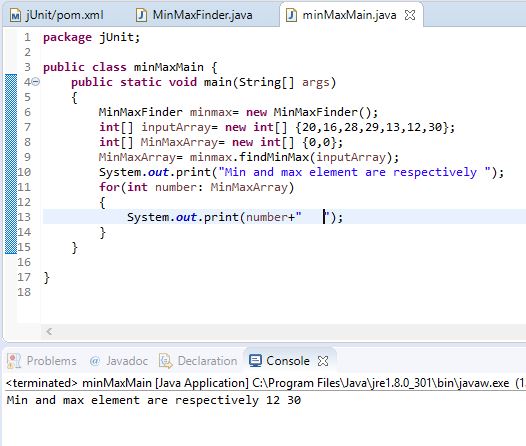
E.g.

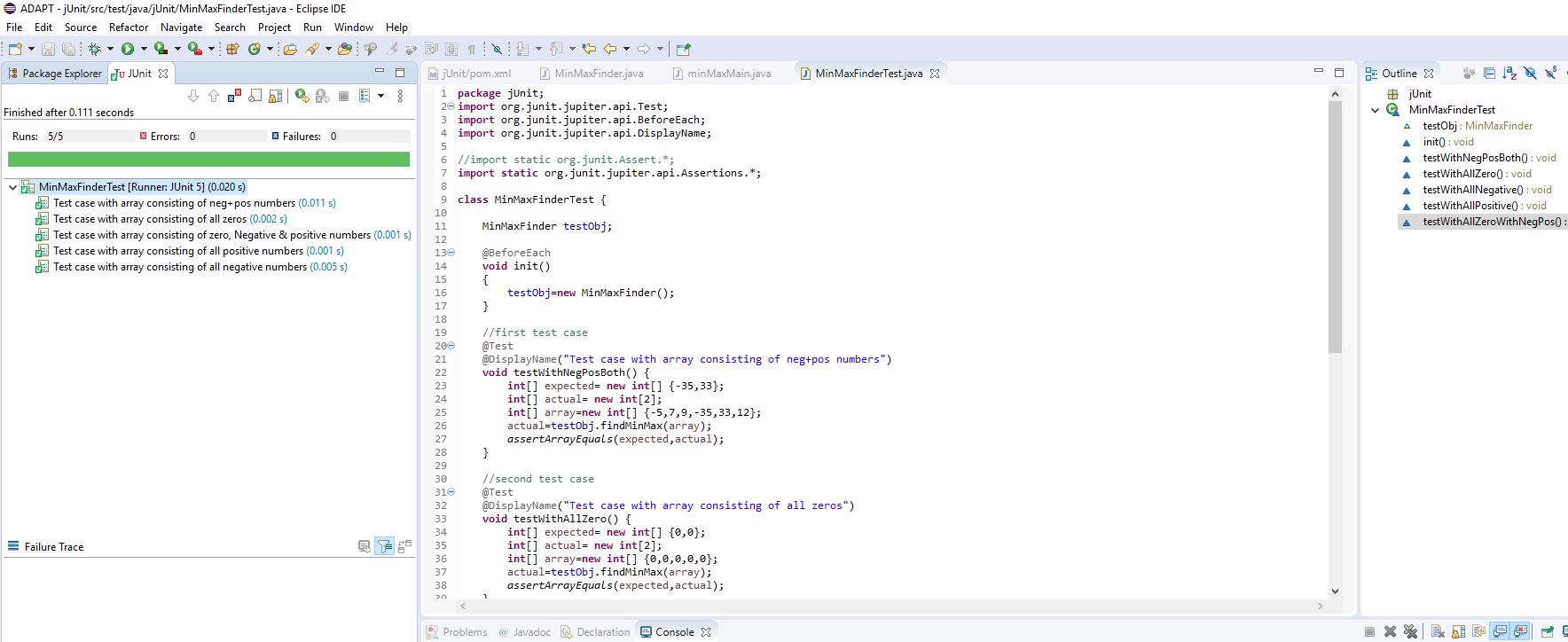
MinMaxFinder.findMinMax( new int[]{56, 34, 7,3, 54, 3, 34, 34, 53} ); should return a new

array with min and max values {3, 56} at 0th and 1st index respectively

Ans:







2) Modify the above method to return a single object representing min and max value of the

pass array. Define new sets of Junit Test cases of this modified method.

3) Write a BankAccount class with method withdraw which accepts amount to be withdrawn

from the account (amount to be deducted from the balance of the account). In case there

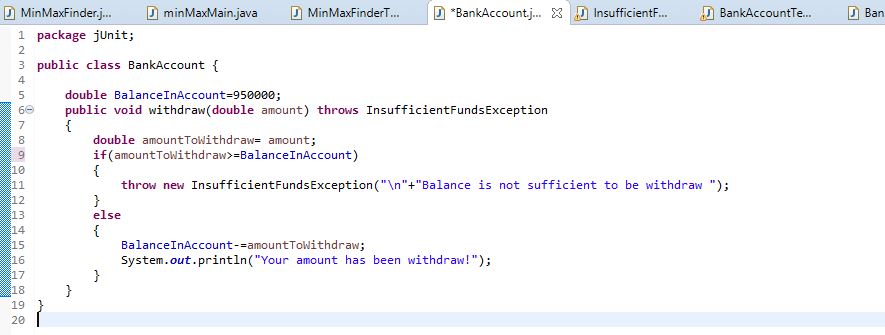
are insufficient funds a InsufficientFundsExpcetion should be raised. After defining the

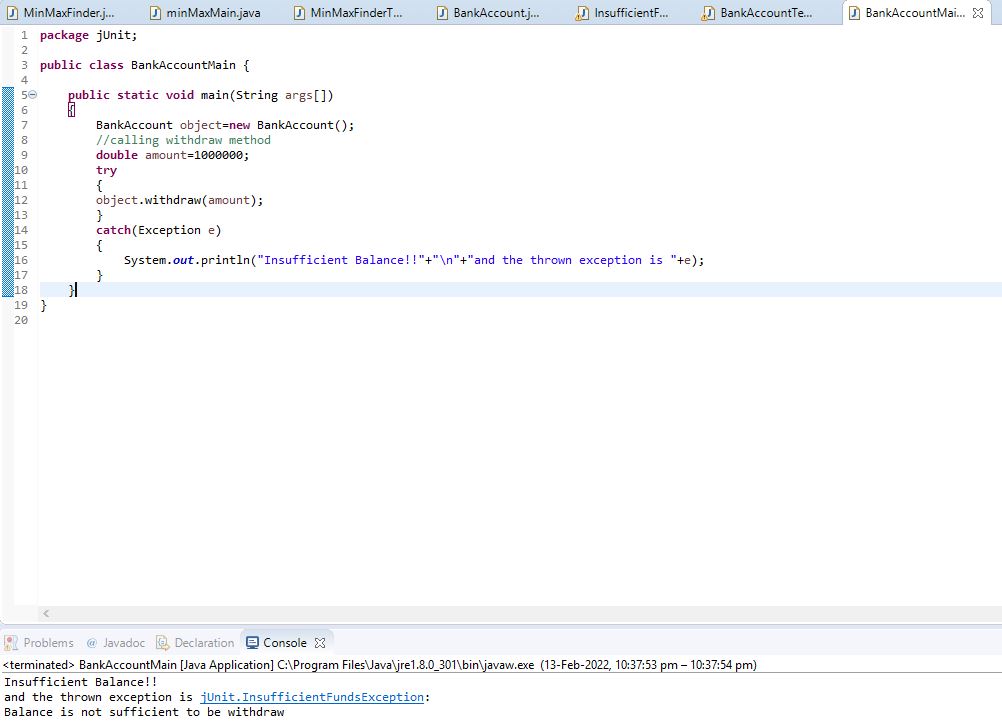
method perform Junit testing to check whether the InsufficientFundsException is raised

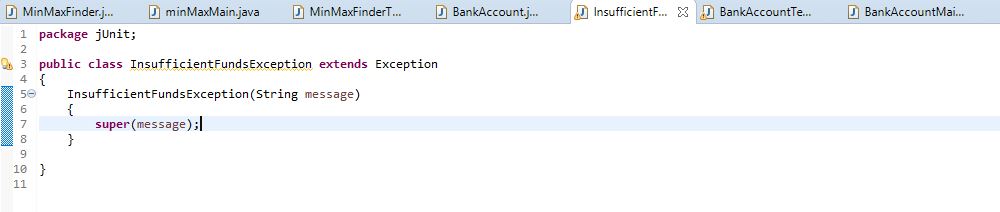
when you try to withdraw amount that is over and above the account balance.

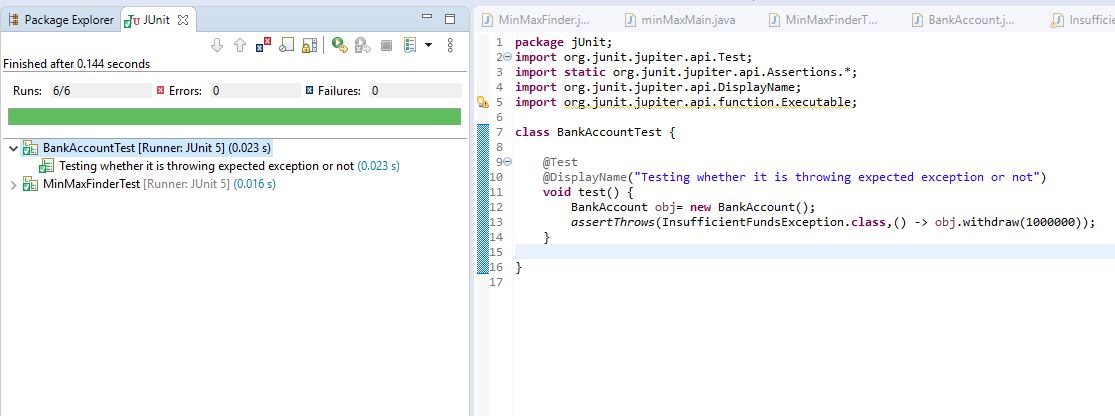
bankAccount.withdraw(20,000); should raise the InsufficientFundsException if the

balance in the account is less than 20,000.



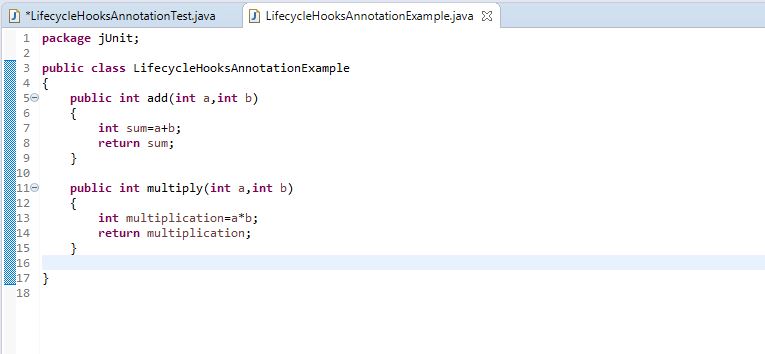


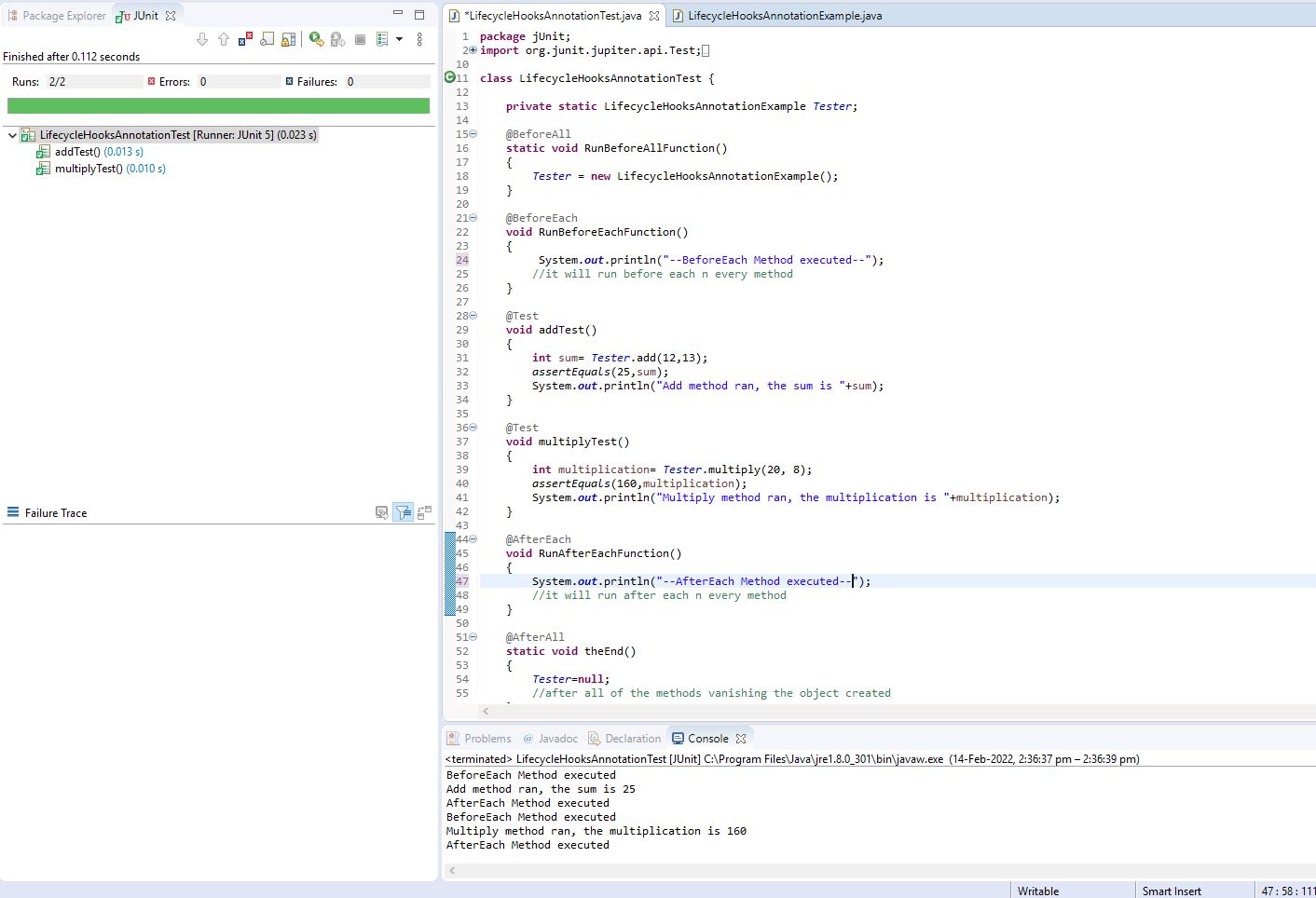




4) Write a Junit Testing to show the use of Lifecycle hooks annotation such as @BeforeAll,

@BeforeEach @AfterEach and @AfterAll

****

****