JUnit automated test suite

@author Anastasiia Ryzhkova Id G00426061

Refactoring

The source code InsuranceProgram.java has been refactored with the aim to make the code suitable for creation and running of the automated test suite.

The following classes have been added:

- InsuranceCalculator.java which is base class for insurance calculation with shared logic for both age groups
- Under25InsuranceCalculator.java which performs calculation for individuals under 25 with specific logic
- Over25InsuranceCalculator.java which performs calculation for individuals 25 and older with specific logic
- o InsuranceConstants.java which contains constants related to insurance calculations
- Runner.java class to run the application

Also the age validation has been added with assumption that valid age range is 16-99, with specific logic for under 25 and 25 and over age groups. Validation ensures that user inputs valid age 16-99 and prompts user in case user enters invalid age (e.g. sixteen or age <16 and >99). The invalid age and blank age are tested for correct exceptions in InsuranceCalculatorTest.class.

Test Suite

Test design has been performed using Boundary Value Analysis and Equivalence Partitioning techniques. The decision tables for test cases below.

No accidents

| Age | # of accidents | Test | Age | # of accidents | Test | | |
|---|----------------|----------------------------|--|----------------|--|--|--|
| 16 | 0 | Under25InsuranceCalculator | 26 | 0 | Over25InsuranceCalculator Test.class Method: | | |
| 17 | 0 | Test.class Method: | 98 | 0 | | | |
| 24 | 0 | testCalculateInsurance | 99 | 0 | testCalculateInsuranceWith NoAccidents | | |
| Decision | | | Decision | | | | |
| Surcharge 100. Total amount to pay 600. | | | No surcharge. Total amount to pay 500. | | | | |

Under 25 age group 1-5 accidents

| Age | # of accidents | Age | # of accidents | Age | # of accidents | Age | # of accidents | Age | # of accidents | Test |
|--|----------------|---|----------------|---|----------------|--|----------------|-----|----------------|-----------------------------|
| 16 | 1 | 16 | 2 | 16 | 3 | 16 | 4 | 16 | 5 | Under25Insura |
| 17 | 1 | 17 | 2 | 17 | 3 | 17 | 4 | 17 | 5 | nceCalculator Test.class |
| 24 | 1 | 24 | 2 | 24 | 3 | 24 | 4 | 24 | 5 | Method: testCalculateIn |
| Decision | | | | | | surance | | | | |
| Surcharge 150. Total amount to pay 650. Surcharge 225. Total amount to pay 725. | | Surcharge 325. Total amount to pay 825. | | Surcharge 475. Total amount to pay 975. | | Surcharge 675. Total amount to pay 1175. | | | | |

Under 25 age group 6 accidents

| | | - |
|------------|----------------|---|
| Age | # of accidents | Test |
| 16 | 6 | InsuranceCalculatorTest.class |
| 17 | 6 | Method: Method: testCalculateInsurance |
| 24 | 6 | |
| Dec | ision | |
| No insurar | nce | |

Over 25 age group 1-5 accidents

| Age | # of accidents | Age | # of accidents | Age | # of accidents | Age | # of accidents | Age | # of accidents | Test |
|---|----------------|-----------|---|-----|---|-----|--|-----|----------------|-----------------------------|
| 25 | 1 | 25 | 2 | 25 | 3 | 25 | 4 | 25 | 5 | Over25Insuranc |
| 26 | 1 | 26 | 2 | 26 | 3 | 26 | 4 | 26 | 5 | eCalculatorTest .class |
| 98 | 1 | 98 | 2 | 98 | 3 | 98 | 4 | 98 | 5 | Method: testCalculateIns |
| 99 | 1 | 99 | 2 | 99 | 3 | 99 | 4 | 99 | 5 | uranceWithAcci dents |
| Decision | | | | | | | | | | |
| Surcharge 50. Total amount to pay 550. Surcharge 125. Total amount to pay 625. | | amount to | Surcharge 225. Total amount to pay 725. | | Surcharge 375. Total amount to pay 875. | | Surcharge 575. Total amount to pay 1075. | | | |

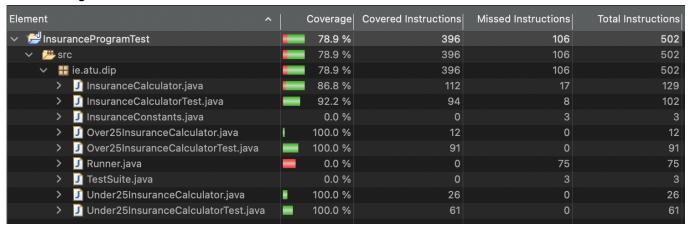
Over 25 age group 6 accidents

| Age | # of accidents | Test |
|--------|----------------|---|
| 25 | 6 | InsuranceCalculatorTest.class |
| 26 | 6 | Method: Method: testCalculateInsurance |
| 98 | 6 | |
| 99 | 6 | |
| | | |
| No ins | surance | |

Test cases for exceptions implemented in InsuranceCalculatorTest.class

Coverage

The coverage is 78.9 %



To Run

Application can be run from Eclipse by Running TestSuite.class or by selecting Coverage as JUnit Test. Test run results from VM below



