Разработка тестов

Задание:

1. Определите области эквивалентности входных параметров.

2. Проведите анализ тестовых случаев, а также учтите граничные значения.

3. Укажите входные значения, покрывающие каждый класс эквивалентности.

4. Выполните тест-кейсы с заданными значениями в главном методе Main.java. Проведите сравнение фактического с ожидаемым.

*Спецификация программы:* Программа для записи студентов (фамилии, имени и возраста), определения минимального и максимального возраста, а также среднего возраста студентов.

Значения возраста должны лежать в диапазоне – [18..50]. Если меньше 18 или больше 50, то присваивается значение 18.

Фамилия и имя должны быть записаны с большой буквы, иначе приводим к нужному регистру.

1. Определяем области эквивалентности для входных параметров.

- область [0:17] – значения возраста не допускающиеся до учебы.

- первая граничное значение 18.

- область [18:50] – значения возраста допускающиеся до учебы.

- второе граничное значение 50.

- область [51:100] – значения возраста не допускающиеся до учебы.

2. Напишем тест-кейсы для каждого класса эквивалентности.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Название сценария | Шаги | Ожидаемый результат | Фактический результат | Результат теста |
| Т1 | Проверка 1 класса эквивалентности | 1.Запустить программу | Программа скомпилируется без ошибок | Программа запускается без ошибок | true |
|  |  | 2.Ввести возраст студента (17) | Так как мы указали значение возраста ниже допустимого, то на консоли должно вывести значение возраста 18 | Программа вывод на консоль значение возраста 18 | true |
| Т2 | Проверка 2 класса эквивалентности | 1.Запустить программу | Программа скомпилируется без ошибок | Программа запускается без ошибок | true |
|  |  | 2.Ввести возраст студента (18) | Так как мы указали значение возраста входящие в диапазон допустимого значения, то на консоли должно вывести введенное нами значения | Программа вывод на консоль введенное нами значение | true |
| T3 | Проверка 3 класса эквивалентности | 1.Запустить программу | Программа скомпилируется без ошибок | Программа запускается без ошибок | true |
|  |  | 2.Ввести возраст студента (32) | Так как мы указали значение возраста входящие в диапазон допустимого значения, то на консоли должно вывести введенное нами значения | Программа вывод на консоль введенное нами значение | true |
| T4 | Проверка 4 класса эквивалентности | 1.Запустить программу | Программа скомпилируется без ошибок | Программа запускается без ошибок | true |
|  |  | 2.Ввести возраст студента (50) | Так как мы указали значение возраста входящие в диапазон допустимого значения, то на консоли должно вывести введенное нами значения | Программа вывод на консоль введенное нами значение | true |
| Т5 | Проверка 5 класса эквивалентности | 1.Запустить программу | Программа скомпилируется без ошибок | Программа запускается без ошибок | true |
|  |  | 2.Ввести возраст студента (51) | Так как мы указали значение возраста выше допустимого, то на консоли должно вывести значение возраста 18 | Программа вывод на консоль значение возраста 18 | true |
| Т6 | Проверка ввода регистра | 1.Запустить программу | Программа скомпилируется без ошибок | Программа запускается без ошибок | true |
|  |  | 2.Ввести имя и фамилию студента начиная с нижнего регистра | Программа должна преобразовать имена и фамилии начинающиеся с нижнего регистра в верхний | Программа выводит в консоль имена и фамилии с верхнего регистра | true |

3. Для проверки тестов будем использовать следующие параметры возраста:

17, 18, 32, 50, 51.

4. Проверка тестов в Eclipse

В ходе работы с кодом, был создан метод TestCase(). В котором создали студентов на основе тест-кейсов, прописанных в 2 пункте. Запустив программу, метод показал, что программа работает корректно и все тесты пройдены успешно.