**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики  
Кафедра информационных систем управления**

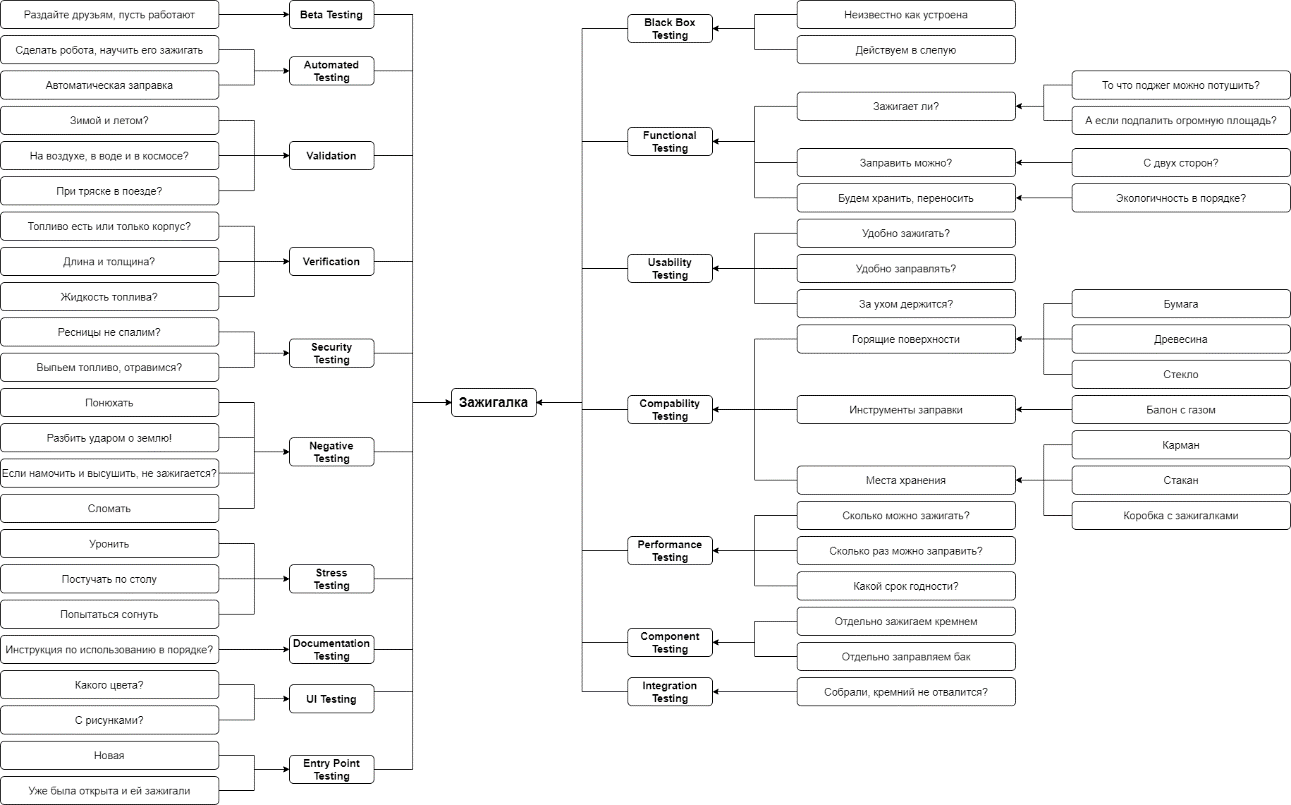
**Отчет  
по тестированию (2 курс)**

Выполнил студент группы №13  
Никончик Даниил Викторович

Минск  
2020

Лабораторная работа №1  
Тестирование предмета

С помощью онлайн сервиса создал схему тестирования выбранного предмета “зажигалка”.



Лабораторная работа №1.2  
Тестирование предмета

Необходимо создать сайт с вопросами по выбранному объекту, в данном случае предмет «Зажигалка». Сайт должен иметь поля для ввода ответов, а также кнопку, при нажатии на которую осуществляется проверка введенных значений согласно заранее созданному текстовому файлу с ответами. Сайт создается при помощи HTML с использованием стилей CSS. Для подключения файла с ответами и осуществления проверки используется JavaScript. Пример сайта – рис 1, файл с ответами – рис 2, код сайта - рис 3.

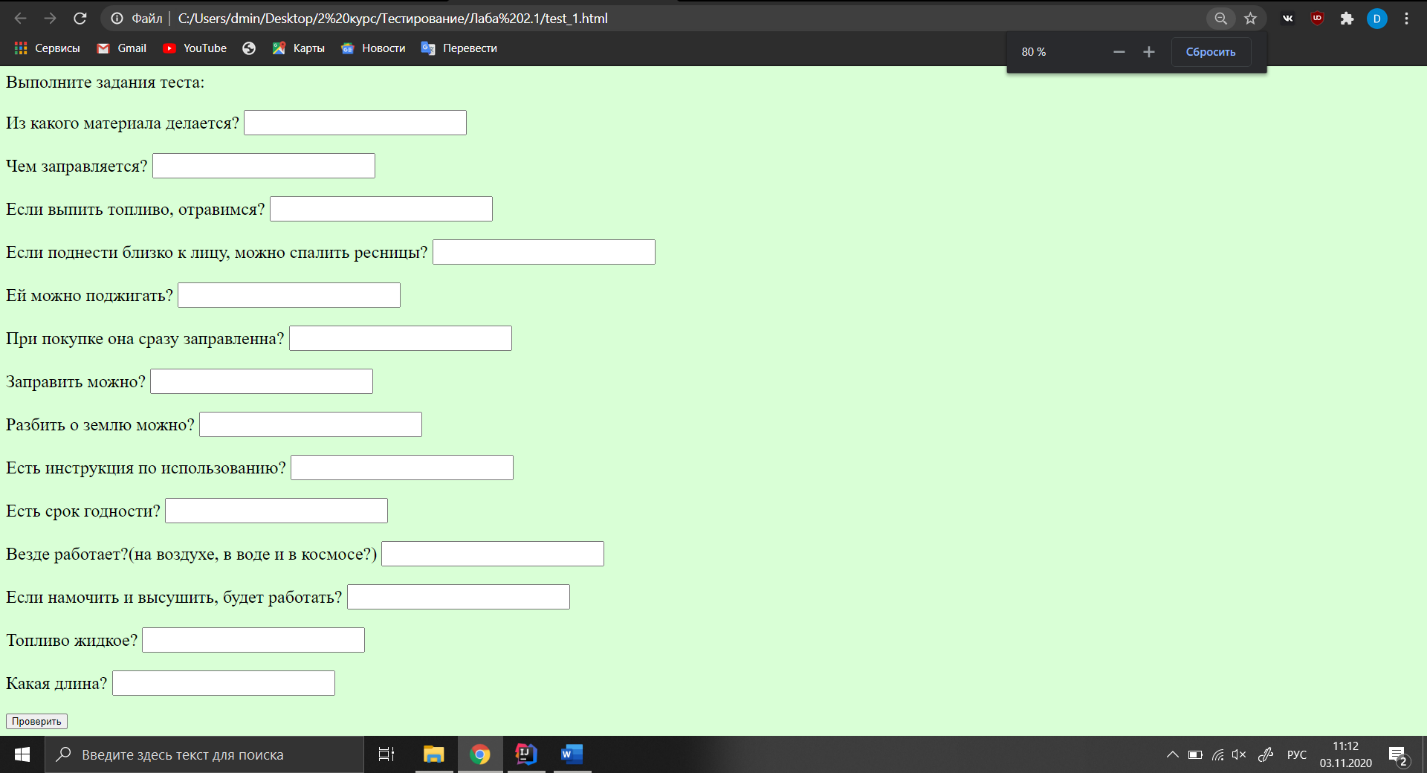


Рис 1

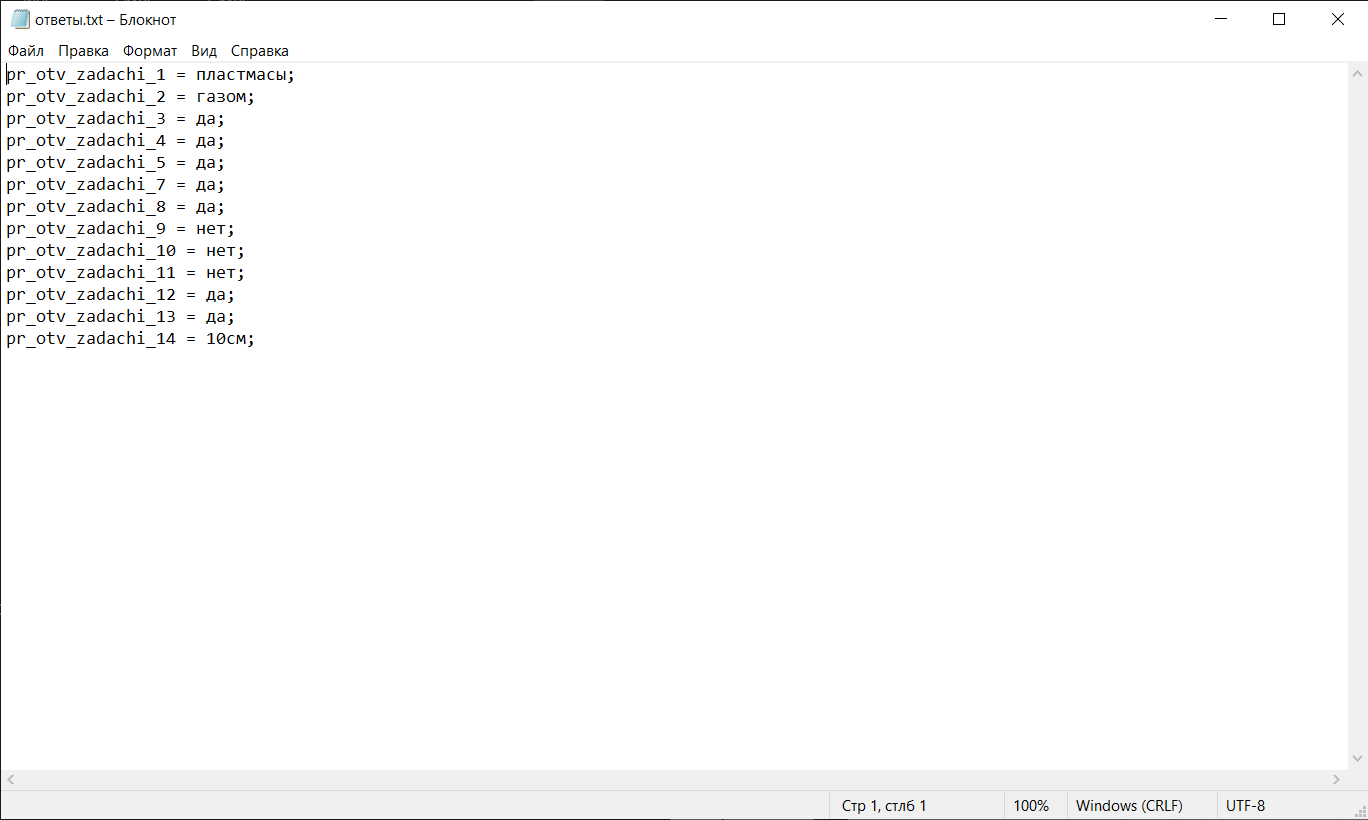


Рис 2

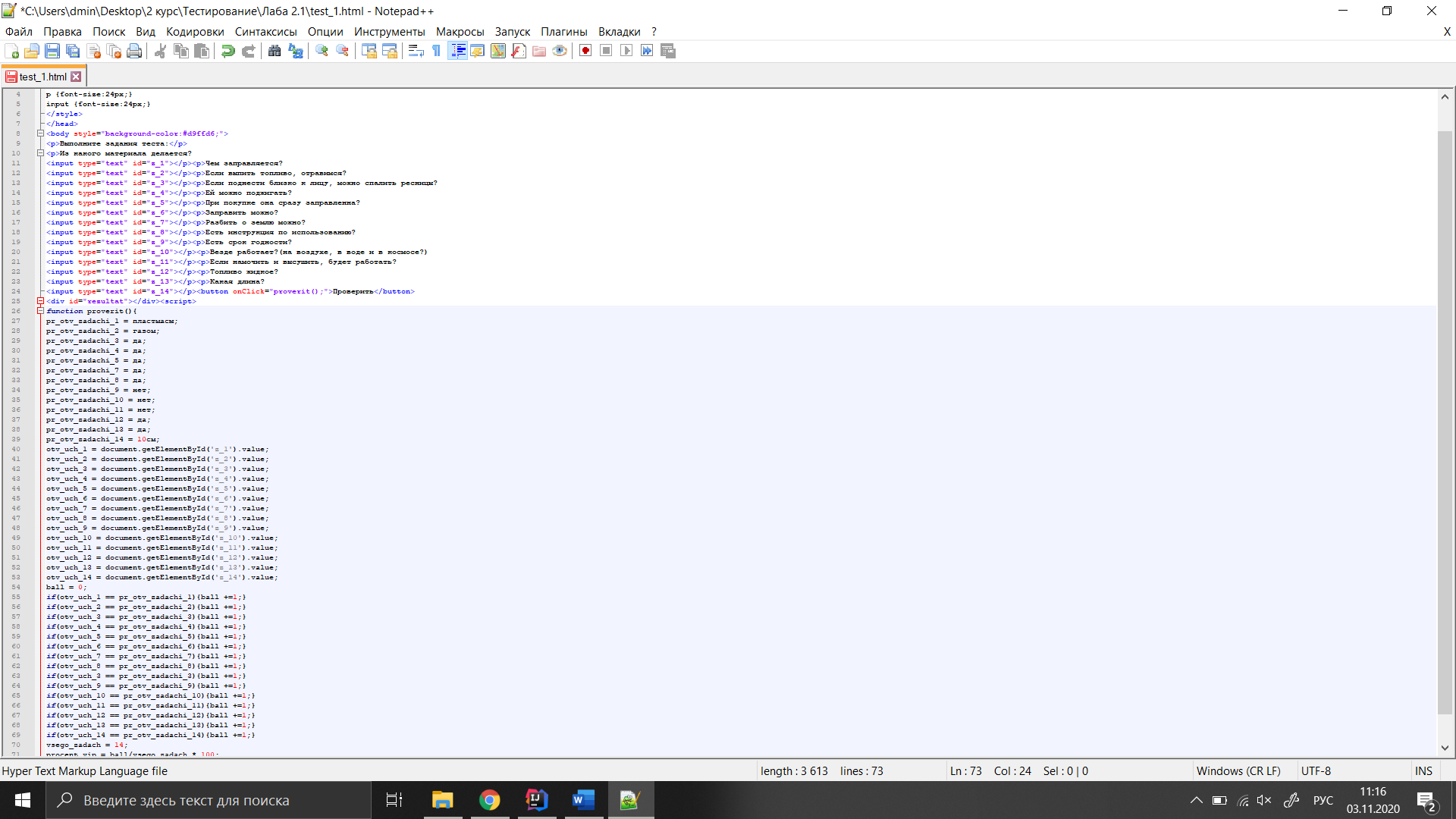


Рис 3

Лабораторная работа №2  
Тестирование треугольника

**Спецификация**

Необходимо разработать программу на языке Java. Программа читает три целочисленных значения (данные могут вводиться с клавиатуры или из файла). Три значения интерпретируются как длины сторон треугольника. На консоль выводится сообщение о том, может ли существовать треугольник с такими сторонами или нет. Рис. 1- код главного класса, рис. 2 – вывод на консоль.



Рис 1

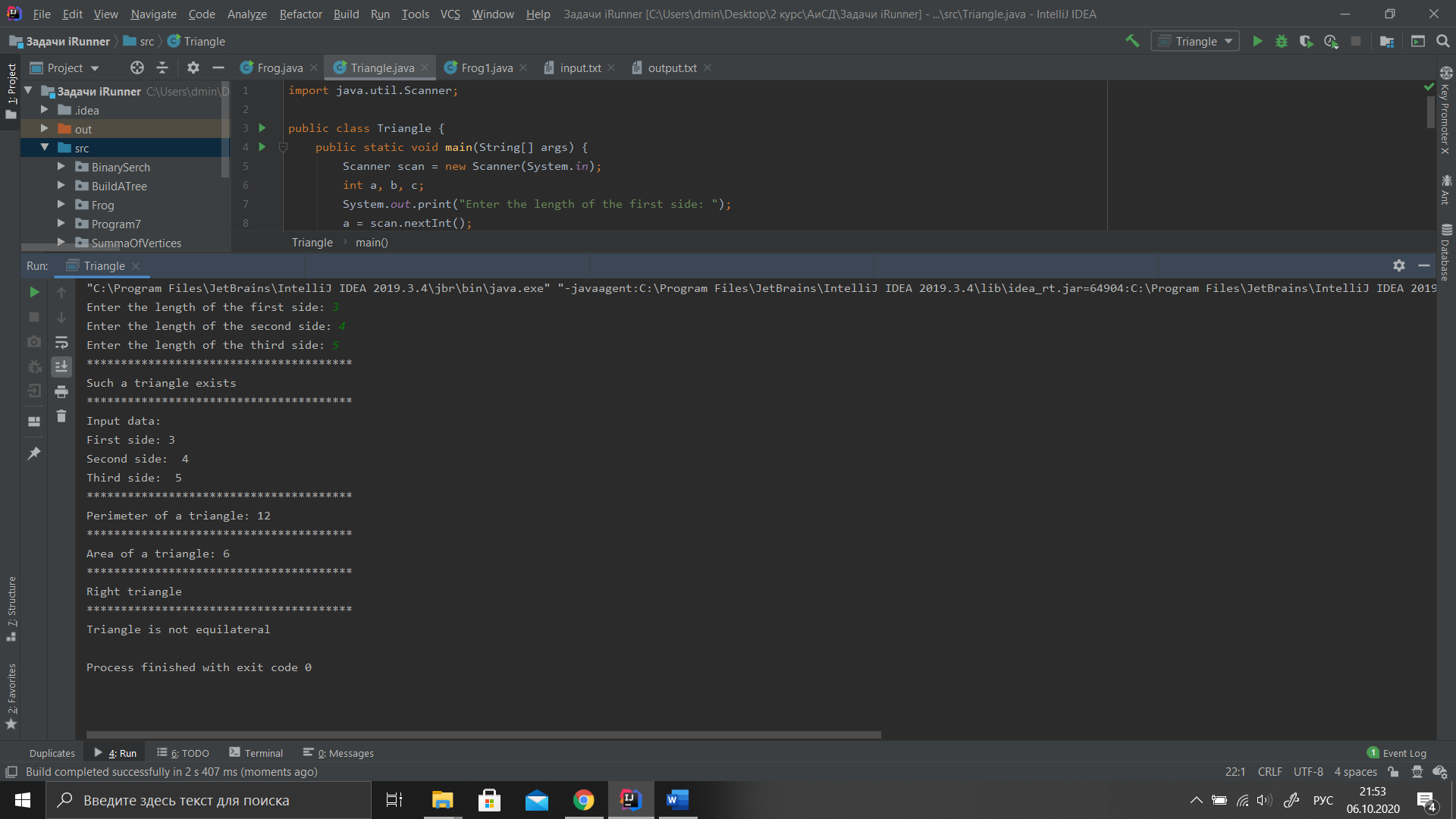


Рис 2

Лабораторная работа №3  
Создание Unit-теста в MS Visual Studio

**Задание:**Было выполнено тестирование проекта, считающего минимальное из трёх чисел с помощью модульного тестирования.

Минимум вычислялся с помощью следующей функции:

Public static int Min(int a, int b, int c)

{

Return Math.Min(a, Math.Min(b, c));

}

Для тестирования был создан проект, содержащий в себе тестирующий класс. Тестирующая функция имеет следующий вид:

[TestMethod]

public void TestMin()

{

int min;

min = Program.Min(3, 4, 5);

Assert.AreEqual(2, min);

}

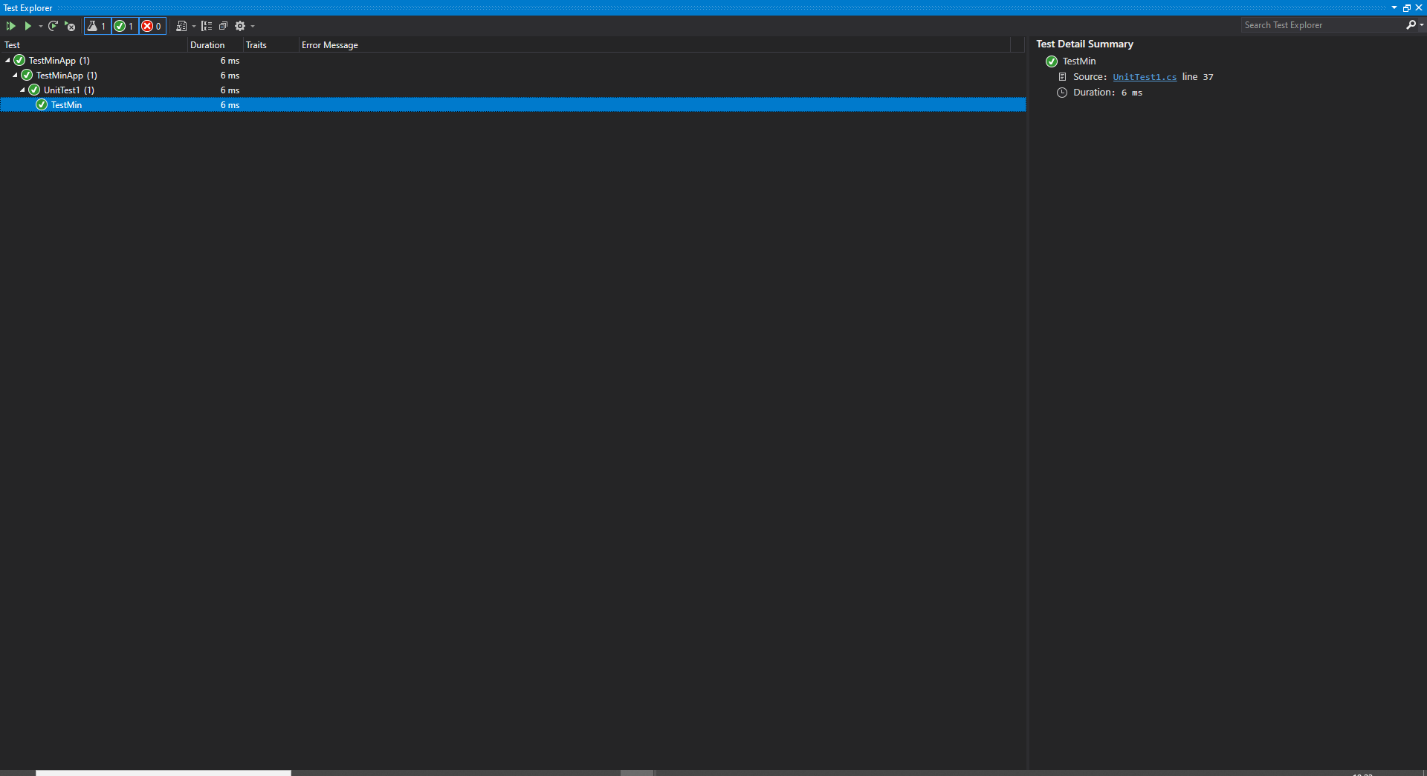
Программа по нахождению минимума трёх чисел была протестирована дважды: на правильном (дано {3, 4, 5}, ответ – 3 (рис. 1)), на неправильном (дано {3, 4, 5}, ответ – 2 (рис. 2)). Результаты выполнения программы отображены ниже на скриншотах. 

Рис. 1.

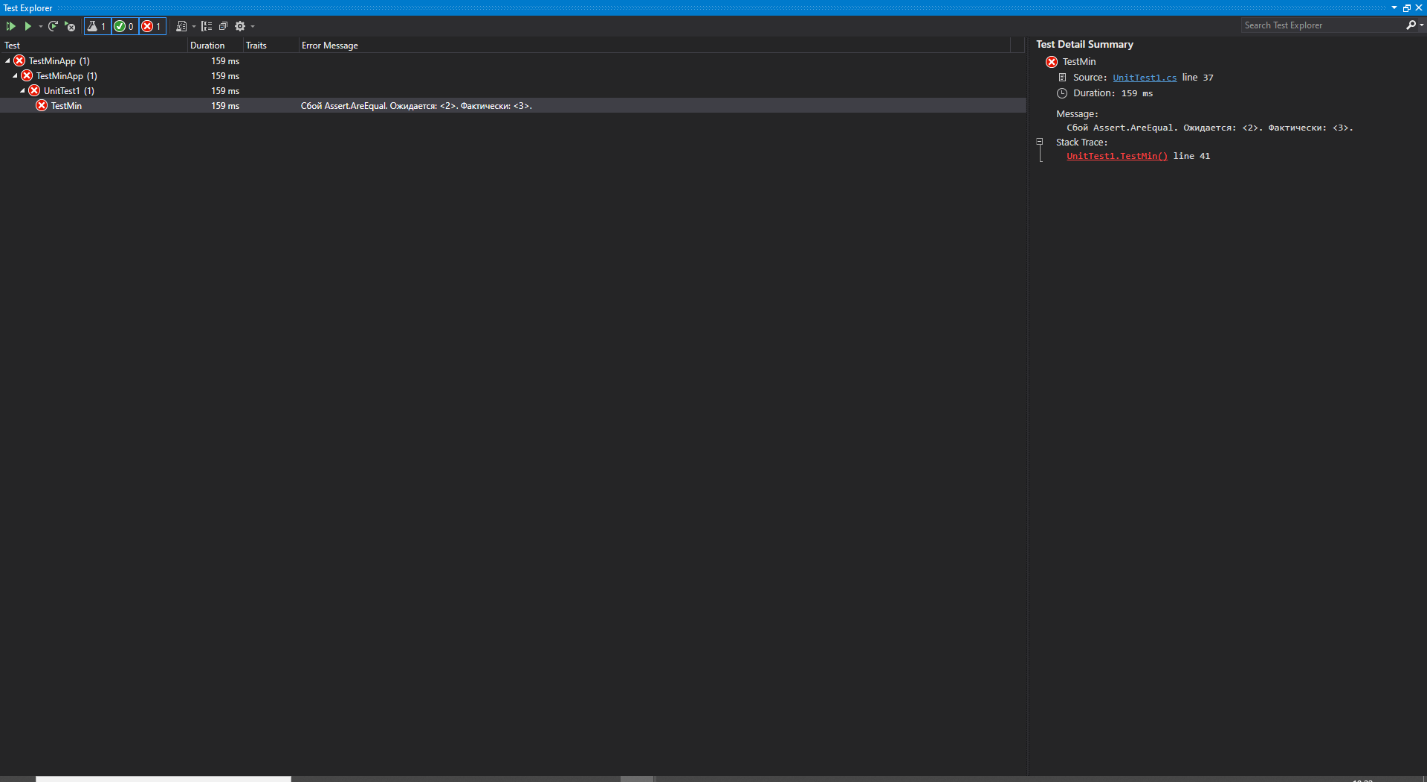


Рис. 2.

Лабораторная работа №4

**Задание:**  
Необходимо протестировать любой сайт при помощи TestComplete. На рисунках 1 и 2 изображены результаты тестирования сайта.



Рис. 1

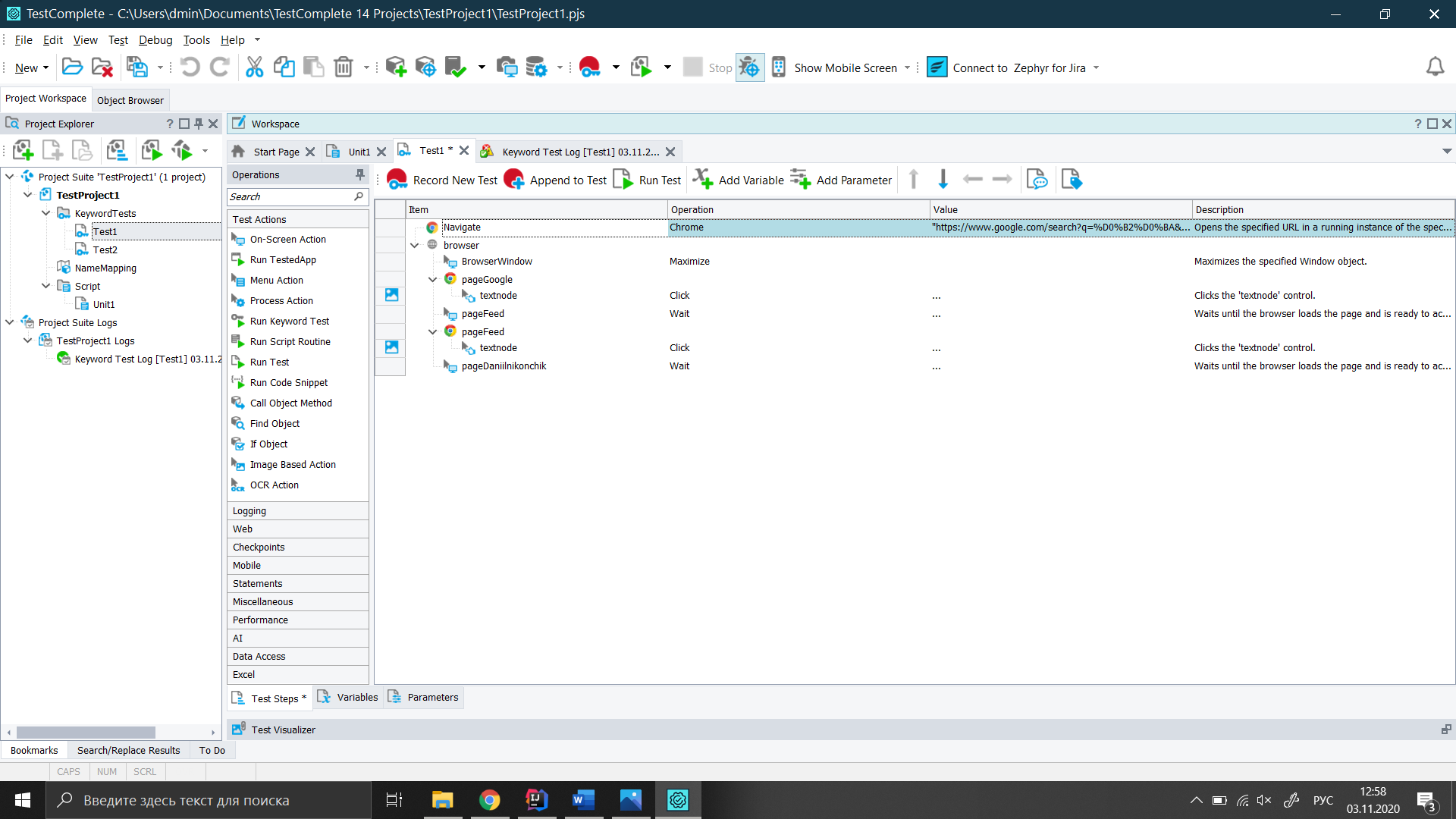
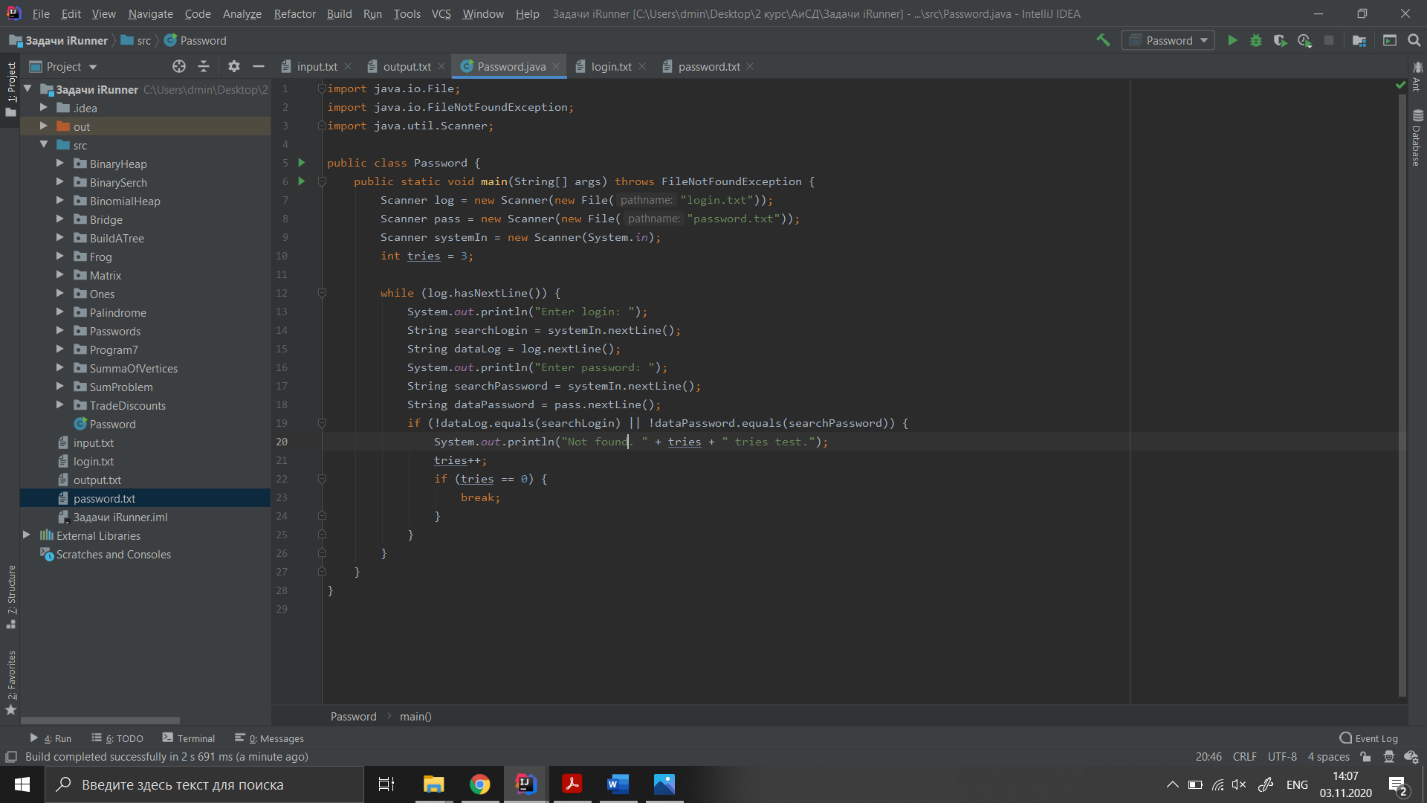


Рис. 2

Лабораторная работа №3-4

Задание 1

Необходимо было реализовать программу которая делает стартовый запрос на ввод логина и пароля с клавиатуры. Пользователь вводит запрашиваемые данные. Программа выполняет поиск введенных данных в файле. Если данные найдены, программа выводит на консоль приветствие или выводит соответствующее сообщение об ошибке. Рис.1 – код программы, рис.2 – вывод в консоли, рис.3 – данные в файле login, рис.4 – данные в файле passwords.

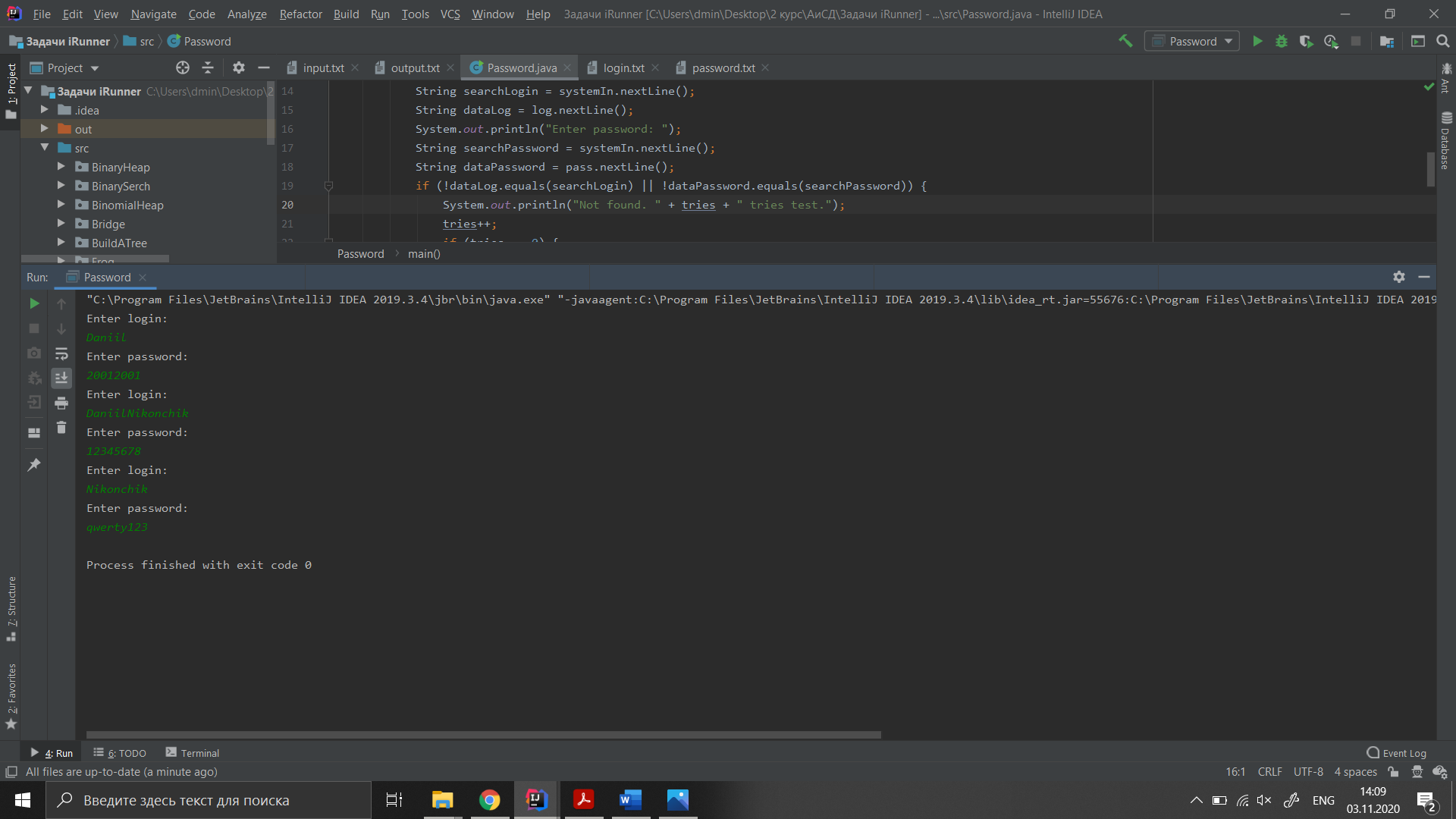
Рис. 1

Рис.2

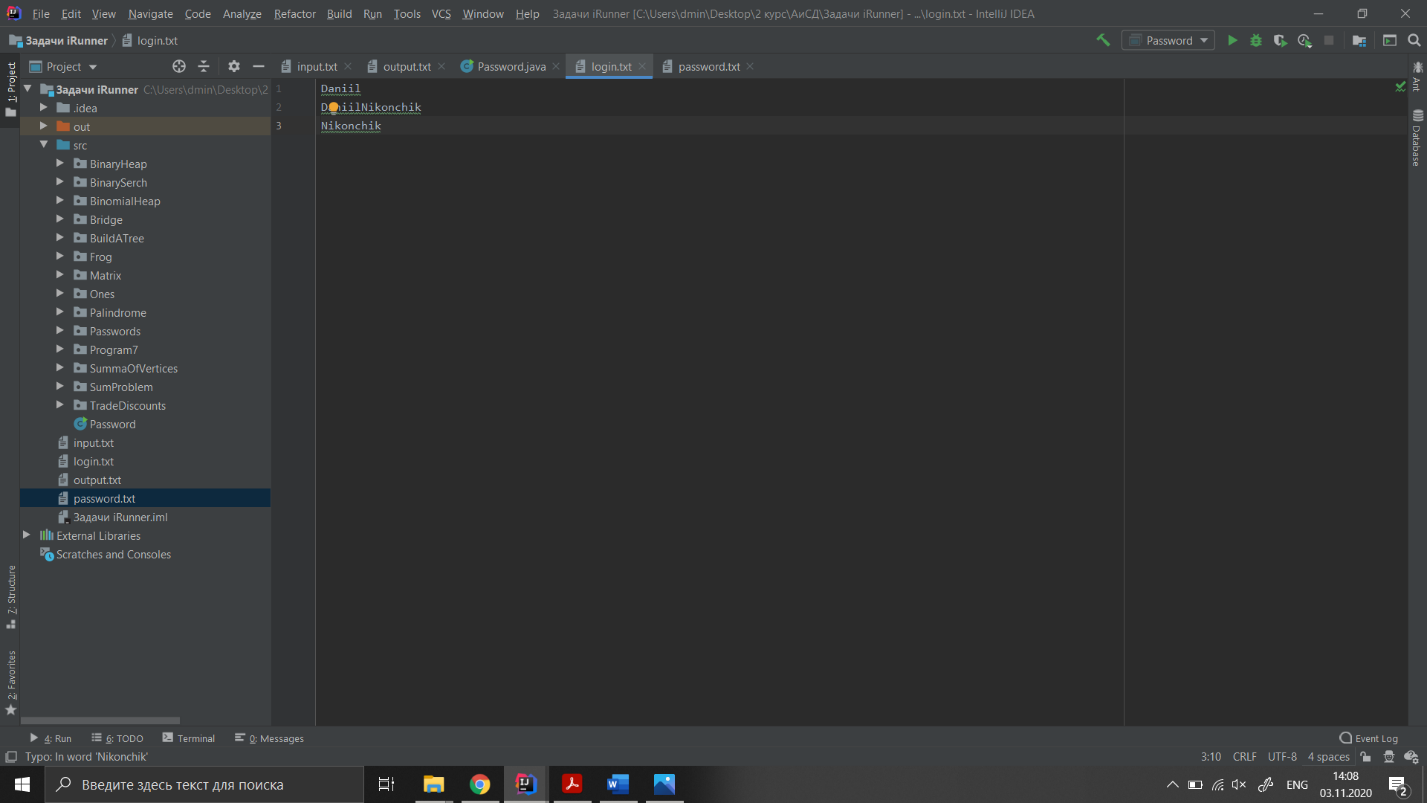
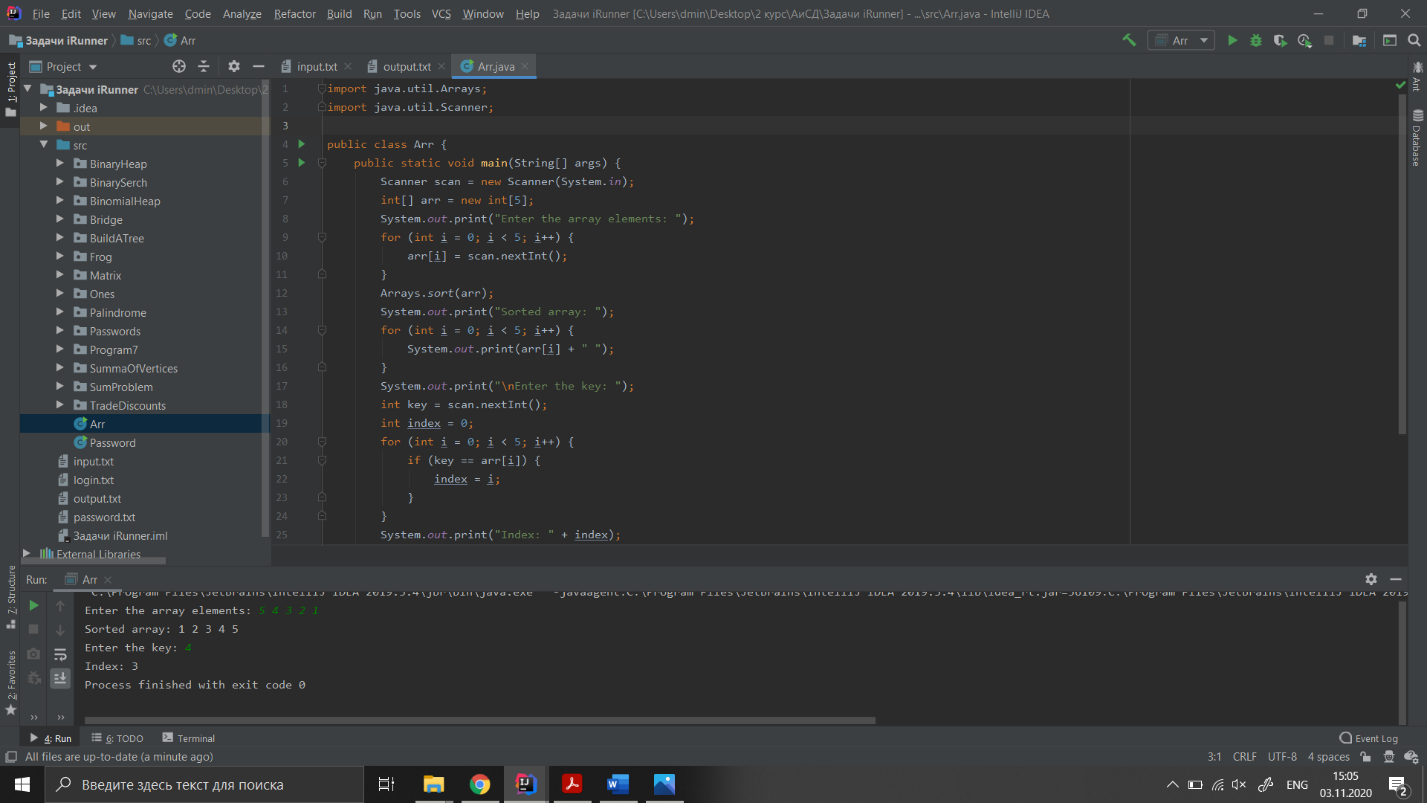
Рис.3

Рис.4

Задание 2

Необходимо было реализовать программу которая выполняет поиск по ключу в упорядоченном массиве M, возвращает индекс элемента массива J, значение которого соответствует ключу поиска Key. Массив и ключ вводятся с клавиатуры. Рис.5 – код программы и вывод в консоли(зеленного цвета то, что вводил самостоятельно)

Рис.5

Лаборатоная работа №5.1

**Цель работы:** разработать рабочую тестовую документацию для тестирования web-приложения.

**Краткие теоретические сведения**

Существует несколько разновидностей рабочей тестовой документации (таблица 1): 1. Check List 2. Acceptance Sheet 3. Test Survey 4. Test Cases Check List – высокоуровневый список проверок, набор правил и критериев, по которым проводится тестирование приложения. Описывает основные проверки для типовой функциональности. Acceptance Sheet - документ, который содержит подробный перечень всех модулей и функций приложения, а также результаты всех тестов данных функций. Как правило, содержит статистику по наиболее важным показателям каждой сборки, определяющим ее качество. Test Survey - документ, который содержит подробный перечень всех модулей и функций приложения, конкретные проверки для них, а также результаты всех тестов. В некоторых случаях для проверок может быть указан ожидаемый результат. Как правило, содержит статистику по наиболее важным показателям каждой сборки, определяющим ее качество. Test Cases (набор тест-кейсов) - набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определенной цели или тестового условия, таких как выполнение определенного пути программы или же для проверки соответствия определенному требованию. Основной фактор выбора тестовой документации – сложность бизнеслогики проекта. Кроме того, определяющими факторами могут быть сроки проекта, размер команды и объем проекта. На одном проекте могут комбинироваться несколько типов тестовой документации. Например, для всего проекта составлен Test Survey, но для наиболее сложных частей составлены тест-кейсы.

**Тестовая документация**

**Единство дизайна**

Дизайн сайта образован популярным сочетанием цветов: белый и синий. От данного сочетания цветов не устают глаза, что является плюсом данного web-приложения. Размер полей данного web-приложения так же является оптимальным.

**Навигация**

Навигация сайта является удобной, интуитивно понятной. Основное меню составлено из текстовых полей с добавленными на них ссылками.

**Функциональность**

Ссылки в данном web-приложении являются работоспособными, использованы в нужных местах. Формы так же являются интуитивно понятными, удобными в использовании. Базы данных в данном приложении визуально не заметны, однако имеют быстрый доступ к необходимым данным. Секретность и анонимность сайта исследовать не удалось. В браузере Google Chrome данный сайт работает хорошо. Фреймы сайта так же работают удовлетворительно. Визуальная, аудио- и видео- наполнение сайта так же удовлетворяют потребностям. Страницы печатаются корректно, баги не наблюдаются. Присутствует форма импорта/экспорта, работает корректно, дизайн приятный. На каждой странице удобные элементы навигации. У каждой страницы наряду с заголовком присутствует подзаголовок выдержанные в одном стиле.

Совместимость с браузером

Web-приложение тестировалось на браузерах Opera, Google Chrome, Microsoft Edge, Yandex. На всех браузерах сайт показал себя с хорошей стороны. Баги не наблюдались. Масштаб везде одинаков.

**Дружественность**

Интерфейс максимально интуитивен, удобочитаем, прост в использовании. Все ссылки являются рабочими, открываются в активной вкладке, что является удобным.

**Работоспособность**

Каждый элемент приложения работает быстро, баги не наблюдаются.

В каждом критерии тестировались следующие элементы:

* Поле
* Кнопка
* Радио баттон
* Чек бокс
* Поле со списком/выпадающий список
* Меню
* Окно
* Скроллинг
* Ссылка
* Таблица
* Поп-ап
* Календарь
* Поле для загрузки файлов
* Сообщения
* Общие проверки

**Выводы по работе**

Тестовая документация составлена успешно, было протестировано web-приложение ВКонтакте. Данное приложение является совершенным, учитывая критерии тестов.

Лаборатоная работа №5.2

Результаты

|  |  |
| --- | --- |
|  | Создание проекта TestLink |
|  | Удаление проекта (сообщение при удаление аналогично для удаления любых составных проекта) |
|  | Прикрепить файлы к плану тестирования |
|  | Инвентарь |
|  | Управление сборкой |
|  | План тестирования |
|  | Тестовая спецификация (фильтр) |
|  | Тестовая спецификация (тестовый проект) |
|  | Набор тестов |
|  | Работа с тестовыми случаями |
|  | Импорт тестовых наборов |
|  | Импорт требований |
|  | Экспорт |
|  | План тестирования |
|  | Поиск |
|  | Ключевые слова |
|  | Документ спецификации испытаний |
|  | Платформы |
|  | Обзор требований |
|  | Назначить выполнение теста |
|  | Вехи |
|  | Настройка выполнения теста |
|  | Выполнение теста |
|  | Отчеты и метрики |
|  | Отчеты |
|  | Графики |
|  | Создать требование |

Лаборатоная работа №5   
Дополнительное задание

Нужно было создать простой тестовый класс и выполнить следующие шаги:

1. Создайте свой простой тестовый класс в формате: [LastName]\_Test.
2. Напишите Конструктор для тест-класса, который должен инициализировать имя текущего тестового метода.
3. Создайте специальные методы для общей инициализации и зачистки (уборки) всех ваших тестовых методов. Допустим, эти методы просто выводят на системный вывод «Sistem.out.println («This is initialization code»)» и т.п.
4. Создайте 3 тестовых метода, которые должны сделать следующее:

* test 1 – выводит что-то на системный вывод;
* test 2 – использует любой метод, который должен пройти;
* test 3 – использует некоторый код, чтобы заставить этот тестовый метод всегда быть неудачным.

1. Создать специальный метод, чтобы организовать все ваши тесты в набор тестов, но таким образом, чтобы ваши тесты будут выполнены в обратном порядке: test3, test2, test1.

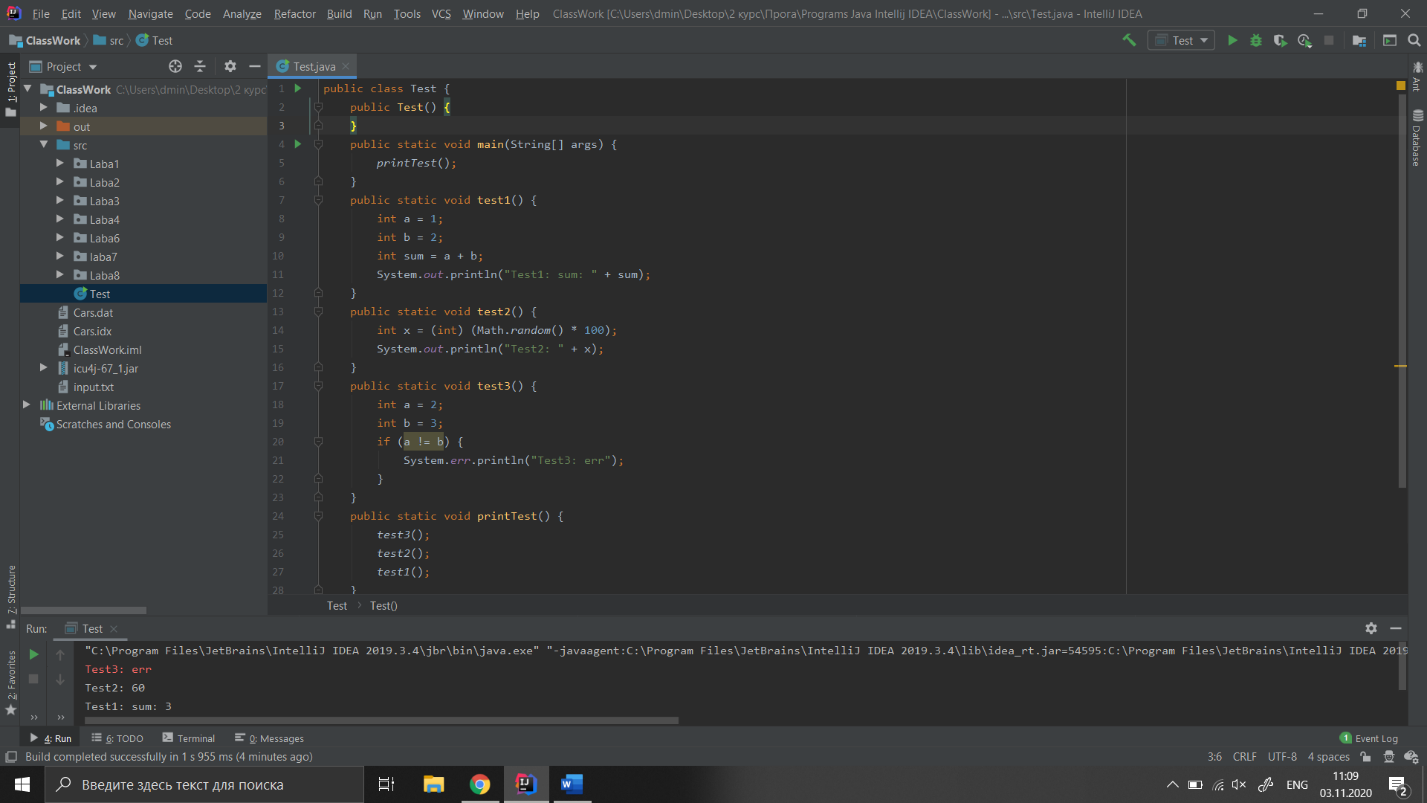
Рис.1 – код класса и демонстрация работы в консоли.

Рис.1

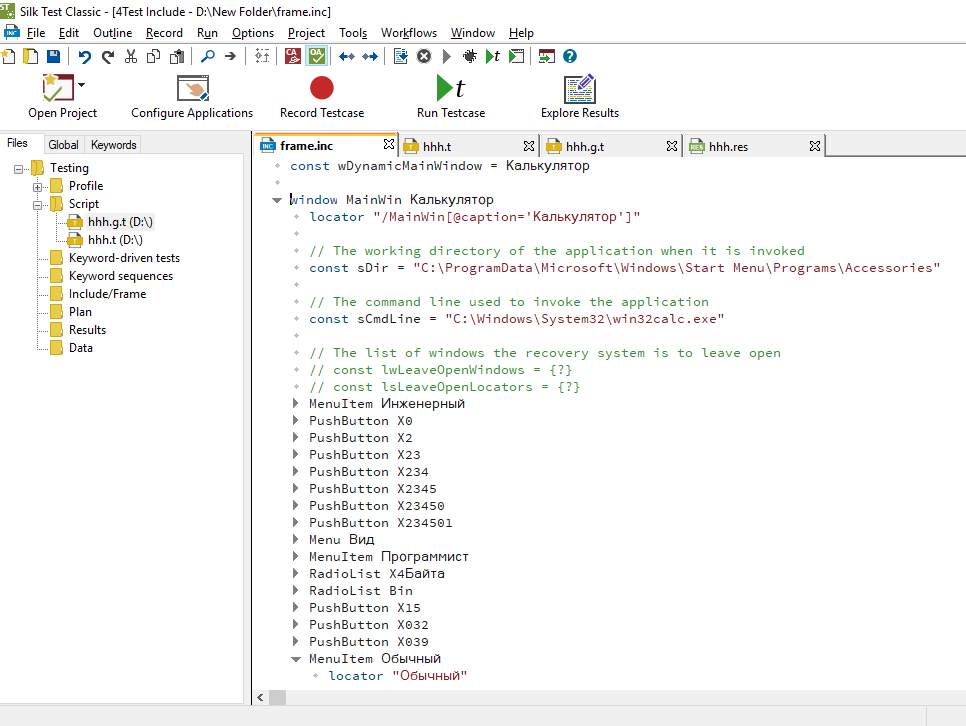
Лабораторная работа №6

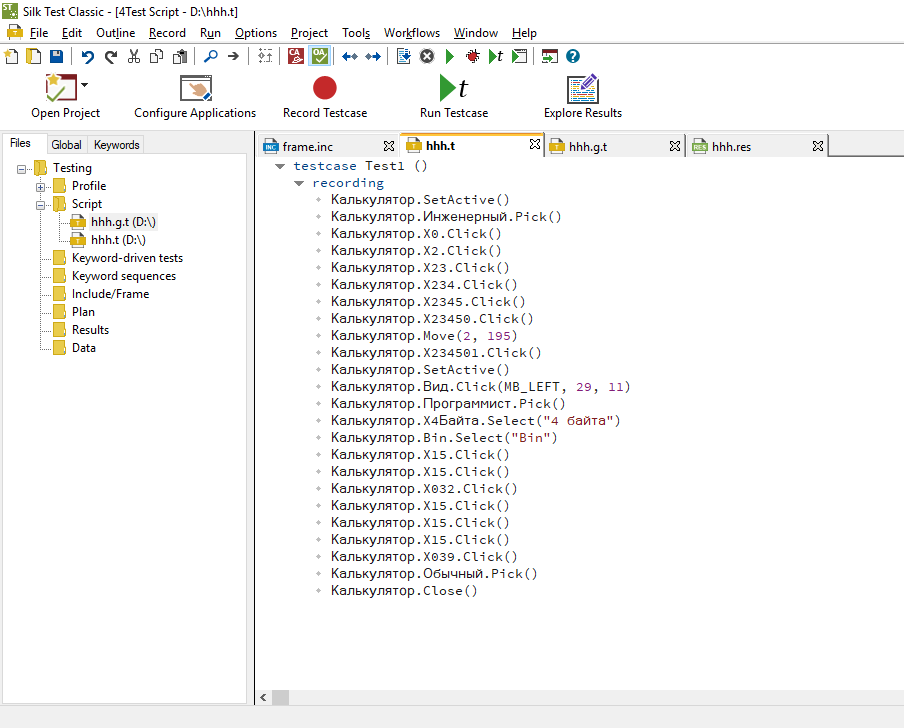
**Задание №3.** Тест-приложение «Калькулятор». Вид «классический».

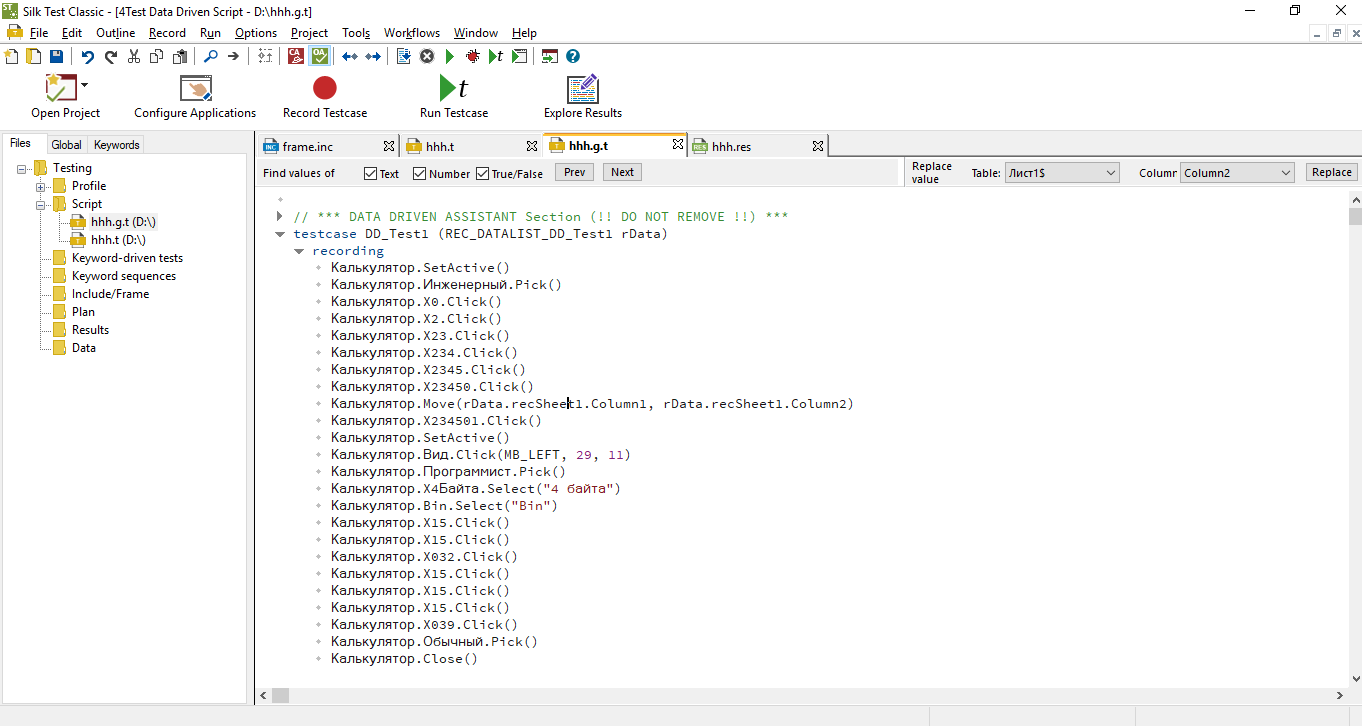
Тестирование функции: инженерные расчеты в 2-ой системе счисления.

1. Выбрать пункт меню «Вид-Инженерный».
2. Проверить доступность кнопок «2», «3», «4», «5», «0», «1» .
3. Получить значения координат из внешних данных (X и Y).
4. Переместить калькулятор по полученным координатам.
5. Текущее значение в поле ввода сбросить в 0.
6. Протестировать выполнение 4-байтных операций умножения в 2-ой системе счисления.
7. Вернуть вид приложения «Вид-Обычный».
8. Закрыть окно «Калькулятор».

Для тестирования использовалась программа SilkTest 20.5







**Вывод:** Silk Test - высококачественная среда тестирования как мелких, так и корпоративных проектов, разработанных в определенном спектре популярных языков программирования, таких как python, C++, Java, С# и т.д., посредством программного тестирования на языке 4Test и уже готовых установленных функций.