**Частное учреждение образования «Минский колледж предпринимательства»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №28**

**по учебному предмету**

*technicaldocs.ru*

*Подпись и дата*

*Взам. инв. №*

*Инв. № дубл.*

*Инв. № подл.*

*Подпись и дата*

**«ТЕХНОЛОГИЯ РАБЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | Выполнили учащиеся 3 курса группы П-2007 |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.В. Рудаковский  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.А. Скриндевский  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.В. Кисель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_М.Ю. Рубченя  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_М.А. Войтеховский |
|  |  |  |
|  |  |  |
|  |  | Преподаватель |
|  |  |  |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.П. Кислюк |

2022

**Цель работы:**

Изучить методы подготовки и проведения тестирования и получить навыки

создания и выполнения тестов для приложений и их компонентов.

**Задание**

1) Модульное тестирование (базовое):

− Открыть исходный код тестируемого приложения.

− Добавить Unit-тест для одной из функций.

− Запустить тест и просмотреть результаты.

− Создать несколько разных тестов (для проверки значений и перехвата

исключений).

1. Тестирование функции IsTheNumberSimple, класс RSA. На Рис.1 мы видим, что тест пройден, число n простое, функция вернула значение true. Функция работает корректно.

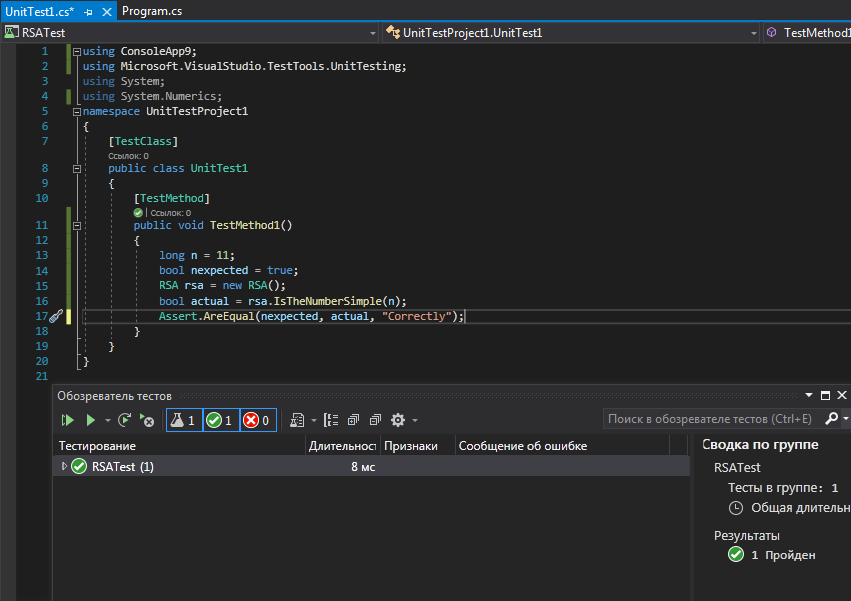


Рис.1. Скриншот из Visual Studio

На Рис.2 мы видим, что тест не пройден, число n не простое, функция вернула значение false. Функция работает корректно.

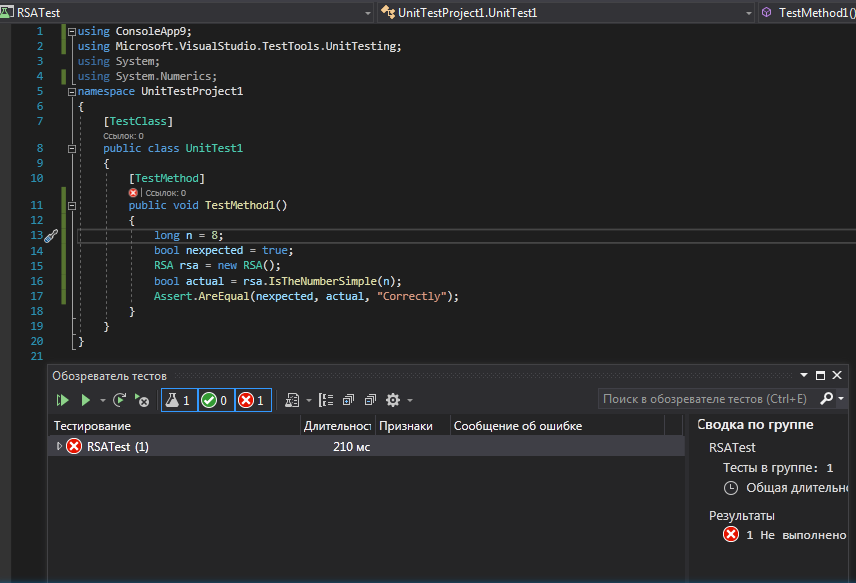


Рис.2. Скриншот из Visual Studio

1. Тестирование функции Calculate\_d, класс RSA.

На Рис.3 мы видим, что тест пройден, функция вернула значение 11. Работа функции заключается в том, чтобы находить такое число d, чтобы было меньше числа m (в данном случае 12) и было взаимно простым с ним (не имело общих делителей, кроме 1). Функция работает корректно

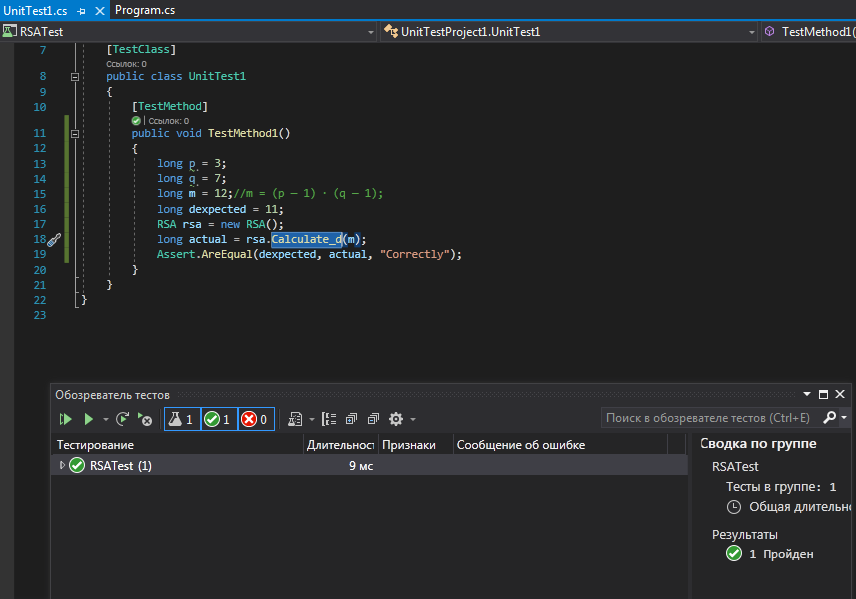


Рис.3. Скриншот из Visual Studio

На Рис.4 мы видим, что тест не пройден функция не вернула значение 10. Работа функции заключается в том, чтобы находить такое число d, чтобы было меньше числа m m (в данном случае 12) и было взаимно простым с ним (не имело общих делителей, кроме 1). Ожидаемое число d = 10, не является числом, которое должна вернуть функция. Функция работает корректно.

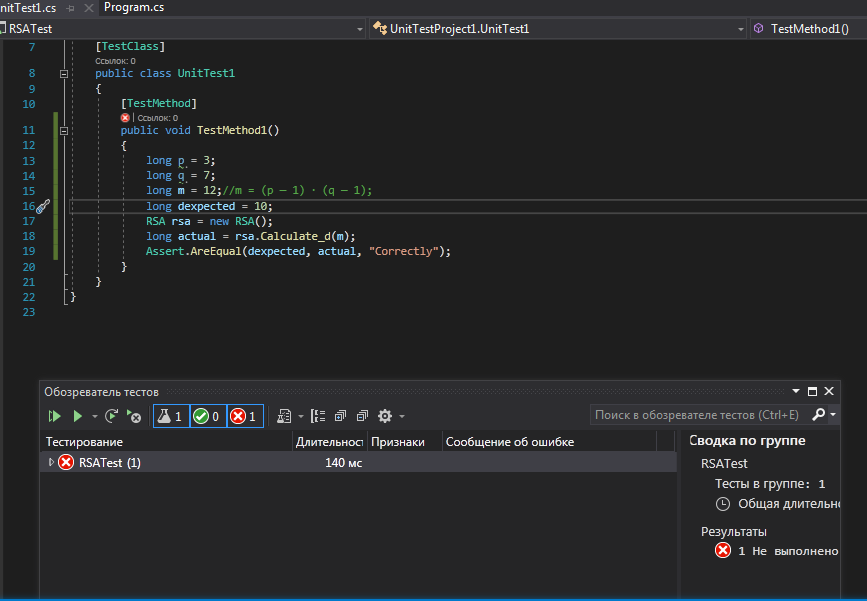


Рис.4. Скриншот из Visual Studio

1. Тестирование функции key, класс Pleifera

Работа функции заключается в том, что она проверяет, подходит ли введенный пользователем ключ.

На Рис.5 мы видим, что тест пройден. Функция вернула значение True, как и ожидалось. Это говорит о том, что ключом могут быть символы русского алфавита, английского, цифры и знаки препинания. Функция работает корректно.

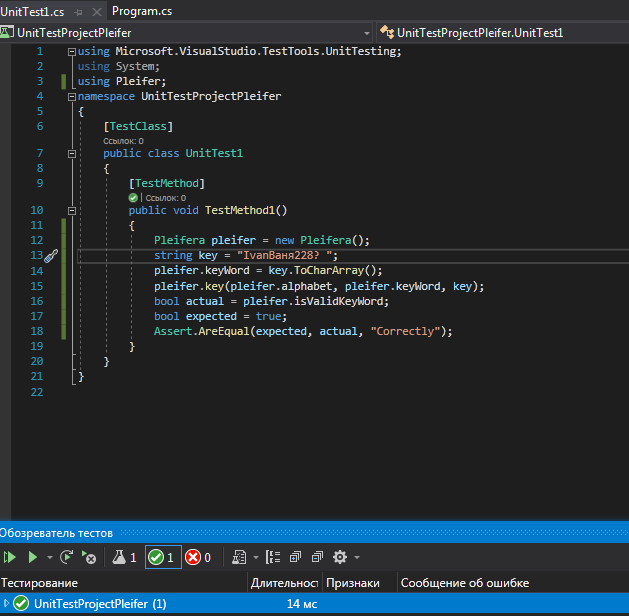


Рис.5. Скриншот из Visual Studio

На Рис.6 мы видим, что тест не пройден. Функция вернула значение False, ожидалось true. Это говорит о том, что ключом не может быть пустая строки. Функция работает корректно.

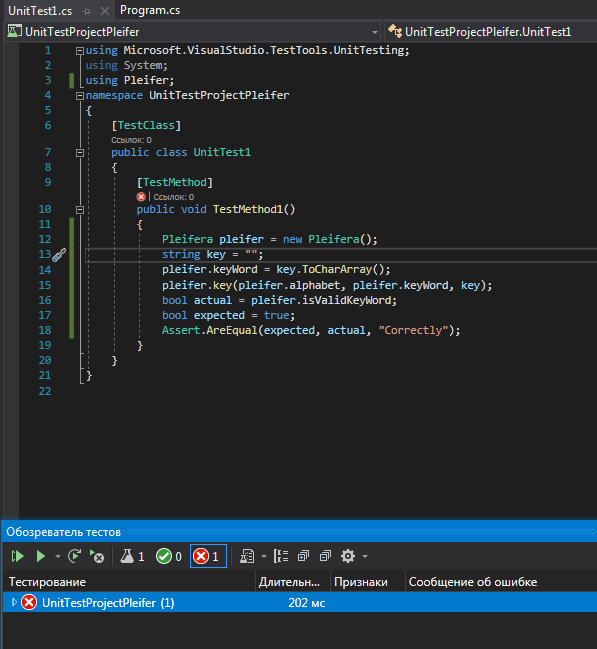


Рис.6. Скриншот из Visual Studio

На Рис.7 мы видим, что тест не пройден. Функция вернула значение False, ожидалось true. Это говорит о том, что ключом не могут быть символы, которых нету в исходном квадрате. Функция работает корректно.

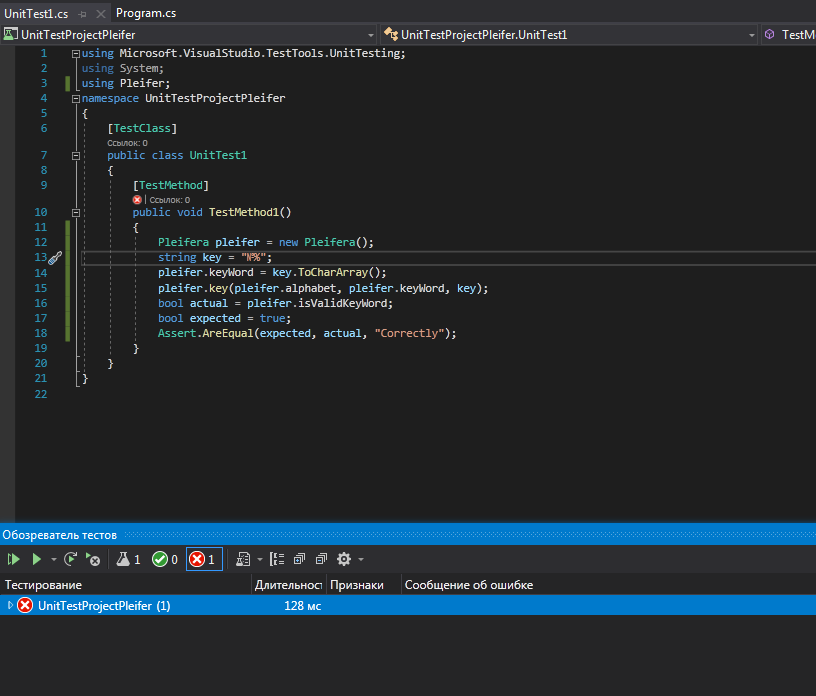


Рис.7. Скриншот из Visual Studio

1. Тестирование функции encrypt, класс Pleifera

Работа функции заключается в шифровании строки по ключу по алгоритму шифрования Полибия.

На Рис.8 мы видим, что тест пройден. Функция вернула строку "ОЛЗЗДЖВЯFLUUFHHЮ$',00T8,", как и ожидалось. Это говорит о том, что функция может шифровать символы русского алфавита, английского, цифры и знаки препинания. Функция работает корректно.

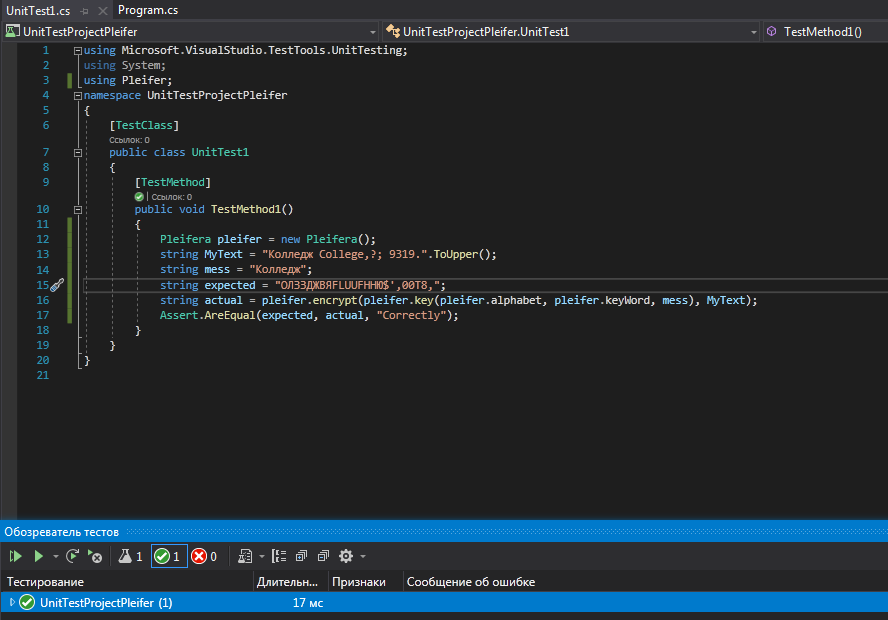


Рис.8. Скриншот из Visual Studio

1. Тестирование функции Decrypt, класс Pleifera.

Работа функции заключается в расшифровании строки по ключу по алгоритму шифрования Полибия.

На Рис.9 мы видим, что тест пройден. Функция вернула строку "ААDR 12., ", как и ожидалось. Это говорит о том, что функция может расшифровать символы русского алфавита, английского, цифры и знаки препинания. Функция работает корректно.

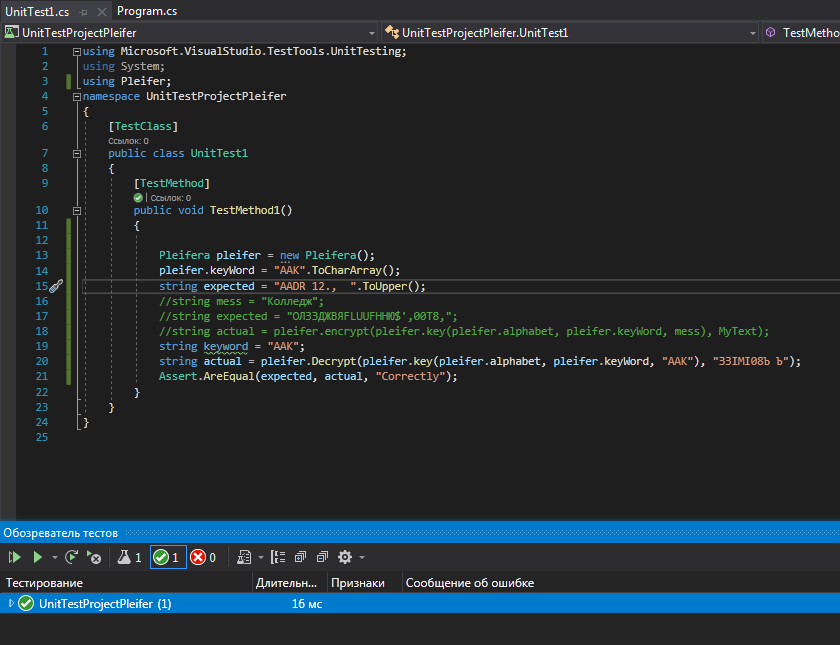


Рис.9. Скриншот из Visual Studio

1. Тестирование функций Decrypt, Encrypt класс RC5

Работа функций Encrypt, Decrypt заключается в шифровании и расшифровке строки по ключу по алгоритму шифрования RC5 соответственно.

На Рис.10 мы видим, что тест пройден. функция Encrypt зашифровала строку «Защита» по ключу, после чего функция Decrypt расшифровала данные и вернула строку «Защита». Функция работает корректно.

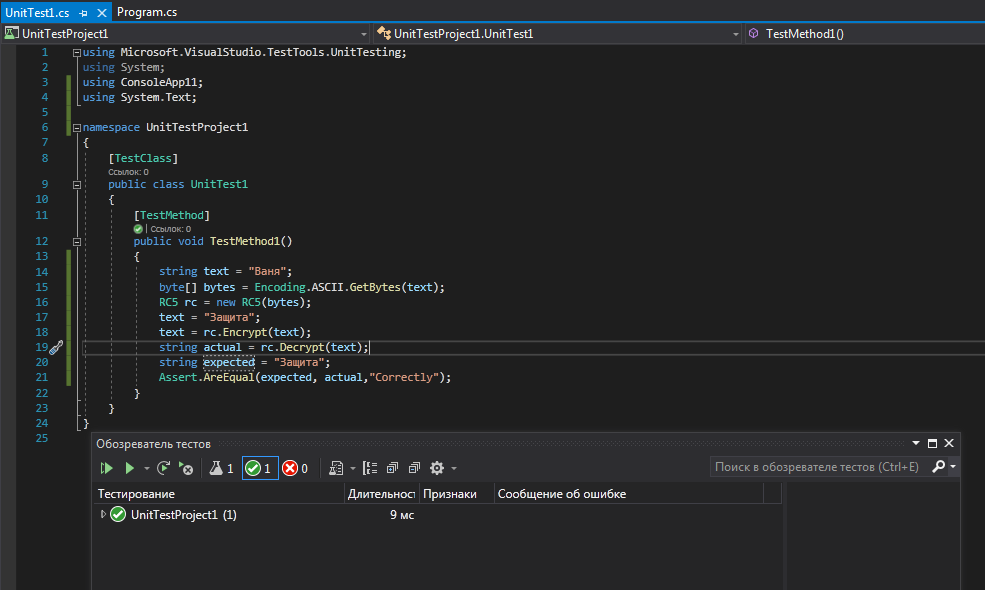


Рис.10. Скриншот из Visual Studio

На Рис.11 мы видим, что тест не пройден, функция Encrypt зашифровала строку «Защита» по ключу, после чего функция Decrypt расшифровала данные и вернула строку «Защита», а не «защита», что говорит о том, что функция корректно шифрует еще и заглавные буквы. Функция работает корректно.

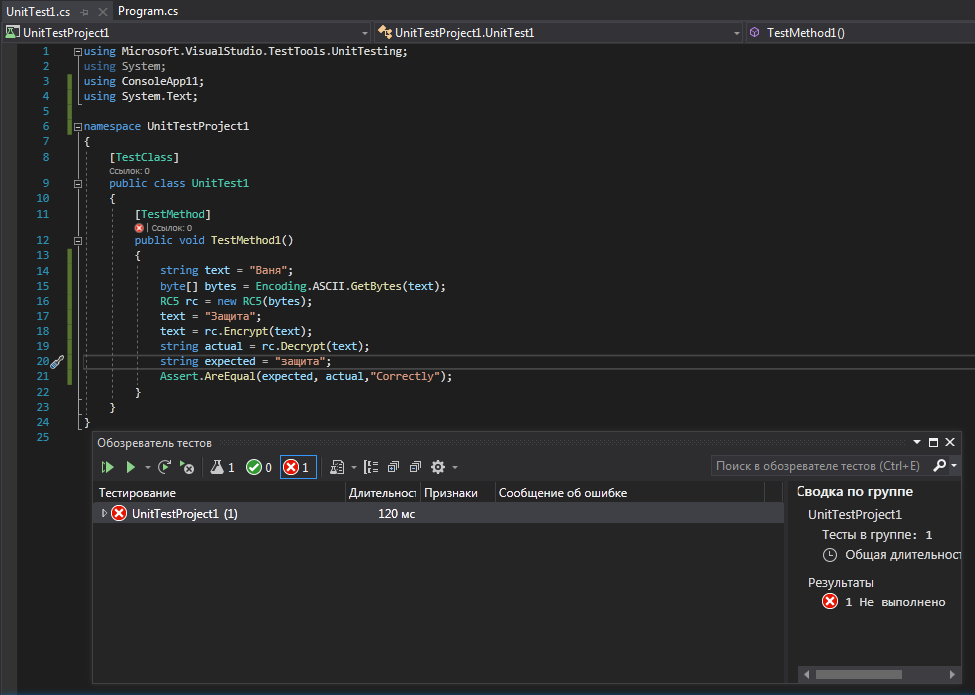


Рис.11. Скриншот из Visual Studio

1. Тестирование функции Encipher, класс BlowFish

Работа функции заключается в шифровании строки по ключу по алгоритму шифрования BlowFish.

На Рис.12 мы видим, что тест пройден, функция Encipher зашифровала строку «Test» по ключу. Функция работает корректно

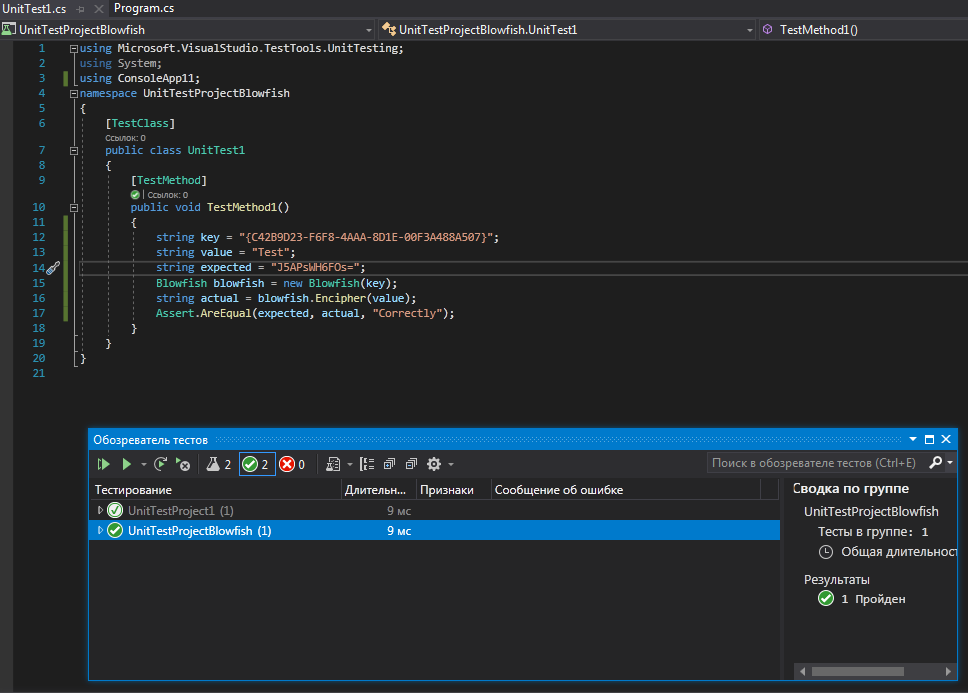


Рис.12. Скриншот из Visual Studio

На Рис.13 мы видим, что тест не пройден, функция Encipher зашифровала строку «test» по ключу. Ожидаемые значения шифрования не подошли, в отличии от предыдущего тестирования. Это говорит о том, что функция корректно шифрует заглавные буквы. Функция работает корректно.

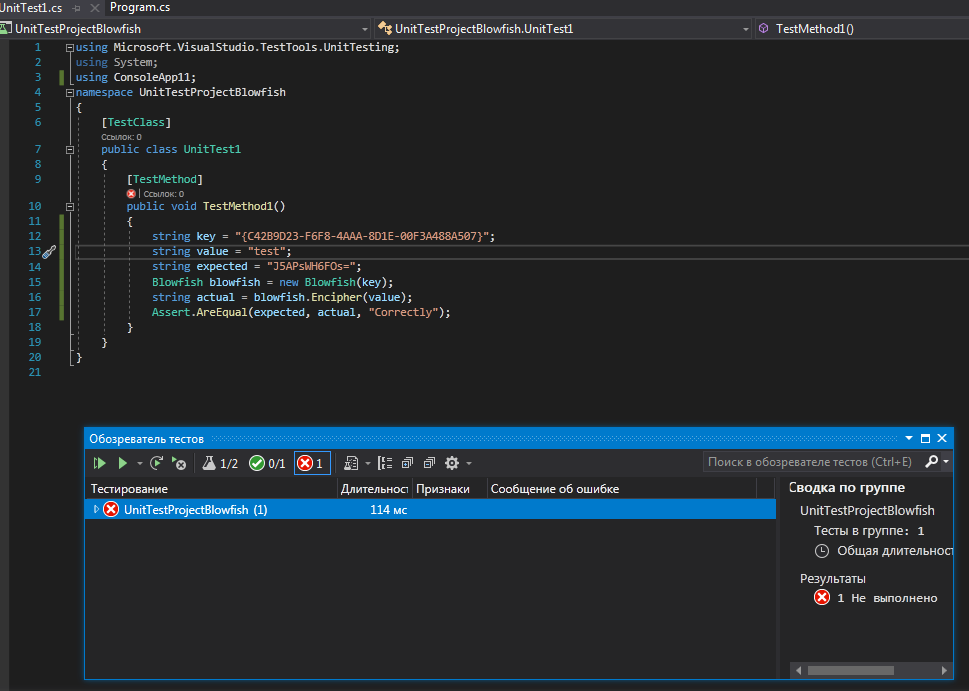


Рис.13. Скриншот из Visual Studio

1. Тестирование функции Decipher, класс BlowFish.

Работа функции заключается в расшифровке строки по ключу по алгоритму шифрования BlowFish.

На Рис.14 мы видим, что тест пройден, функция Decipher зашифровала строку «Test» по ключу, функция Encipher вернула такую же строку «Test». Функция работает корректно.

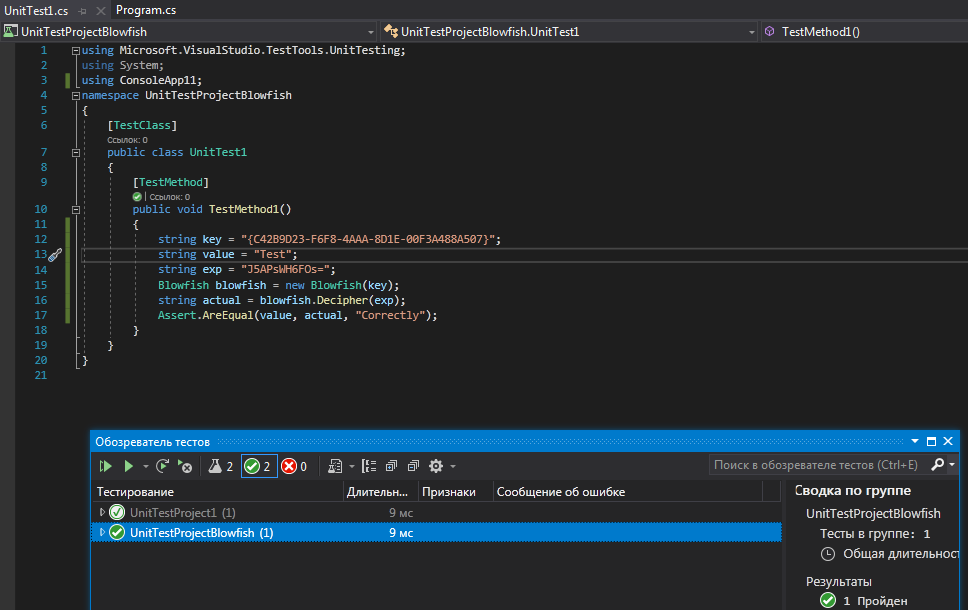


Рис.14. Скриншот из Visual Studio

1. Тестирование функции Encrypt, класс AES

Работа функции заключается в шифровании строки по ключу по алгоритму шифрования AES.

На Рис.15 мы видим, что тест пройден, функция Encrypt зашифровала строку «Бэтмен and Batman» по ключу и вернула ожидаемое значение. Это говорит о том, что функция шифрует символы как русские, там английские символы, к тому же она шифрует еще заглавные буквы. Функция работает корректно.

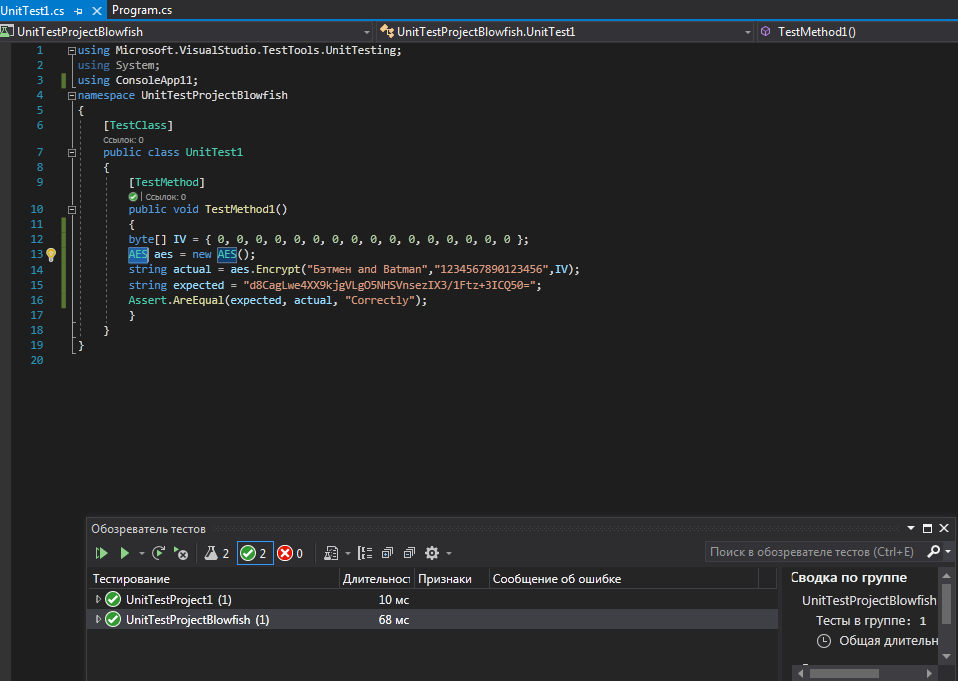


Рис.15. Скриншот из Visual Studio

1. Тестирование функции Decrypt, класс AES

Работа функции заключается в шифровании строки по ключу по

алгоритму шифрования AES. На Рис.16 мы видим, что тест пройден, функция

Decrypt вернула строку «Бэтмен and Batman» как и ожидалось. Функция

работает корректно.

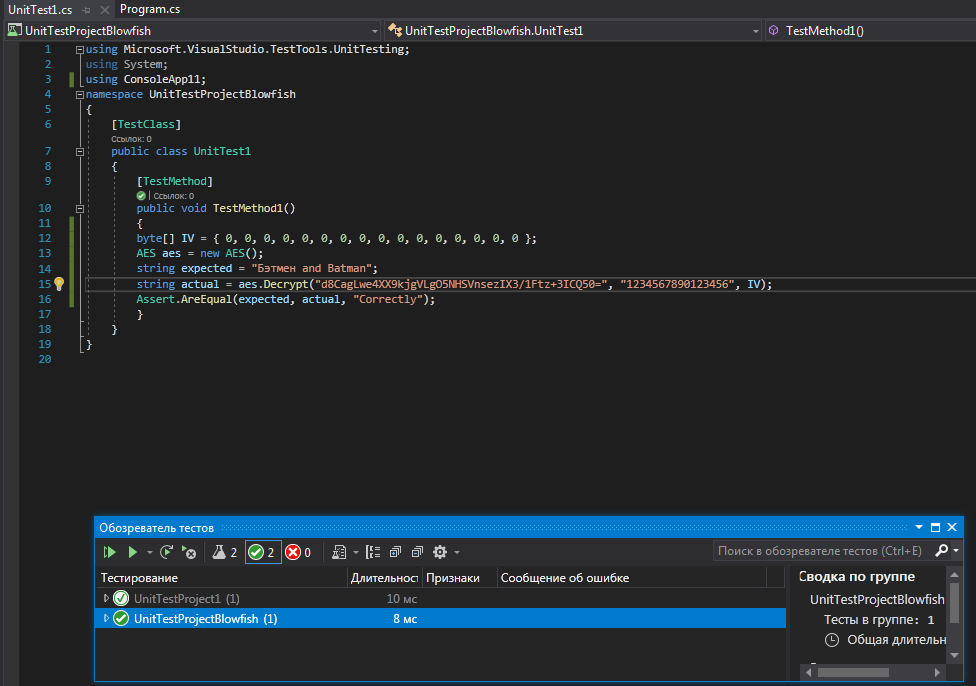


Рис.16. Скриншот из Visual Studio

1. Тестирование функций Decrypt, Encrypt класс XOR.

Работа функций заключается в том, чтобы шифровать и дешифровать

строку исходного текста по ключу по алгоритму шифрования XOR. Ввиду

особенностей алгоритма шифрования тестирование данных функций

отдельно друг от друга не представляется возможным.

На Рис.17 мы видим, что тест пройден, функция Decrypt зашифровала

строку "ВаняIvan 123" по ключу «000» вернула зашифрованное значение,

после чего функция Encrypt расшифровала его в строку "ВаняIvan 123", как и

ожидалось. Это говорит о том, что функции могут шифровать как большие и

маленькие символы английского, так русского и цифры. Функции работают

корректно.

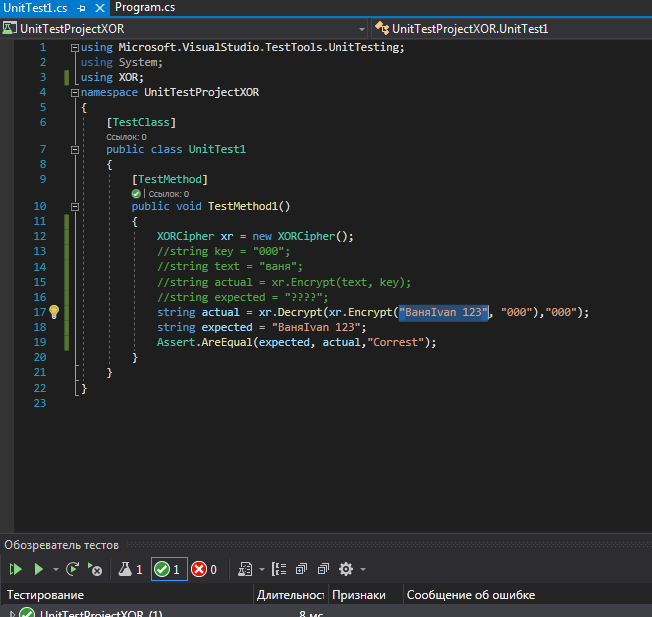


Рис.17. Скриншот из Visual Studio

На Рис.18 мы видим, что тест пройден, функция Decrypt зашифровала строку "ВаняIvan 123" по ключу «роллтон» вернула зашифрованное значение, после чего функция Encrypt расшифровала его в строку "ВаняIvan 123", как и ожидалось. Это говорит о том, что ключом могу являться как буквы, так и цифры. Функции работают корректно.

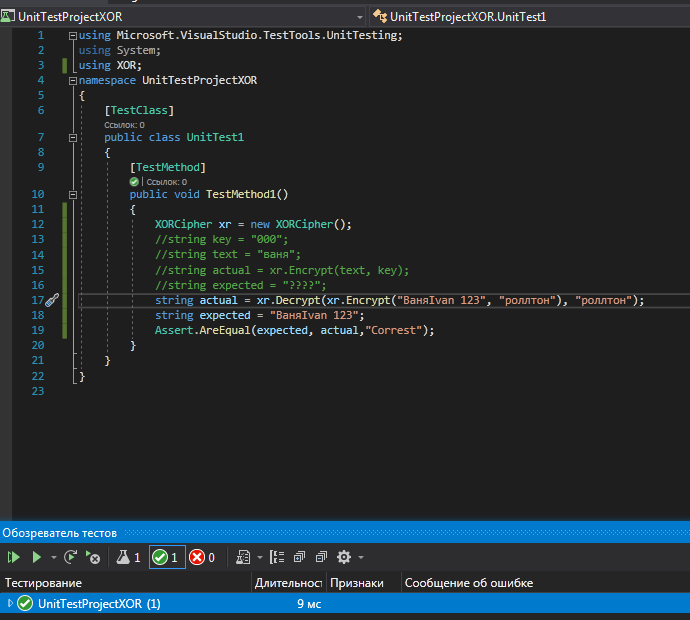


Рис.18. Скриншот из Visual Studio

1. Тестирование функции Encrypt класс VigenereCipher.

Работа функции заключается в шифровании строки по ключу по

алгоритму шифрования Vigenere.

На Рис.19 мы видим, что тест пройден, функция Encrypt вернула строку

" SKPCUCI" как и ожидалось. Функция работает корректно.

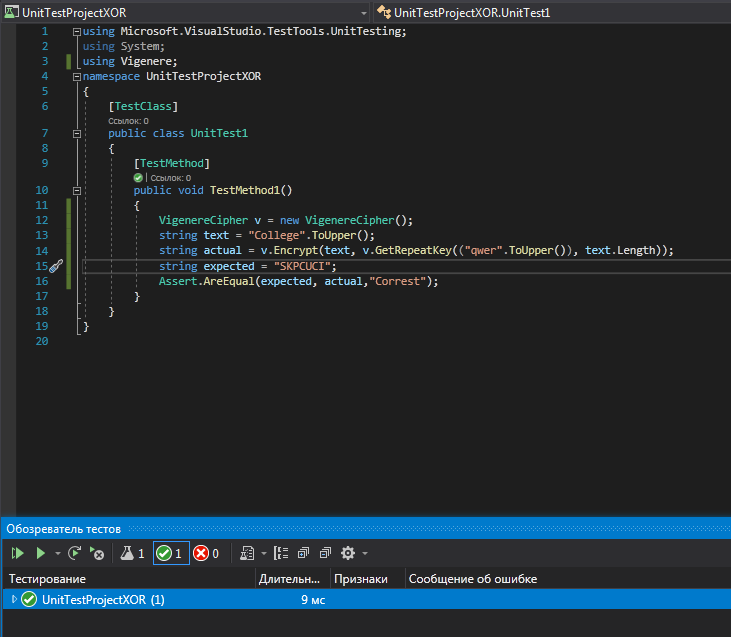


Рис.19. Скриншот из Visual Studio

1. Тестирование функций GetRepeatKey класс VigenereCipher.

Работа функции заключается в подгоне ключа под размер строки

текста для шифрования.

На Рис.20 мы видим, что тест пройден, функции Encrypt вернула

строку "SKPCXEY" как и ожидалось. Это говорит о том, что несмотря на то,

что длина ключа больше длины строки текста для шифрования, она

корректно подогнала ее под размер строки. Также можно отметить, что в

предыдущем тесте, функция корректно подогнала размер ключа под размер

текста для шифрования, в то время как длина ключа была меньше строки для

шифрования. Функция работает корректно

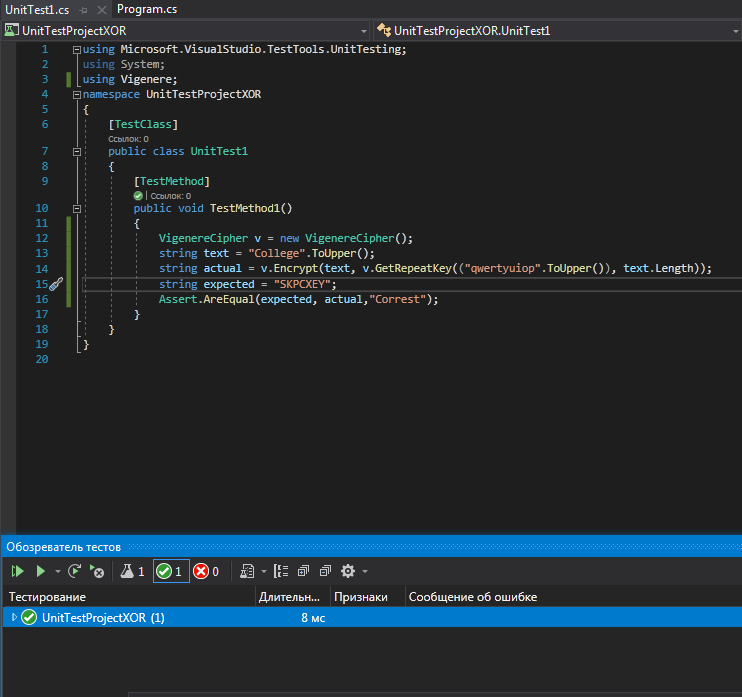


Рис.20. Скриншот из Visual Studio

1. Тестирование функции Encrypt класс VigenereCipher.

Работа функции заключается в расшифровке строки по ключу по

алгоритму шифрования Vigenere. На Рис.21 мы видим, что тест пройден,

функция Encrypt вернула строку " COLLEGE" как и ожидалось. Функция

работает корректно.

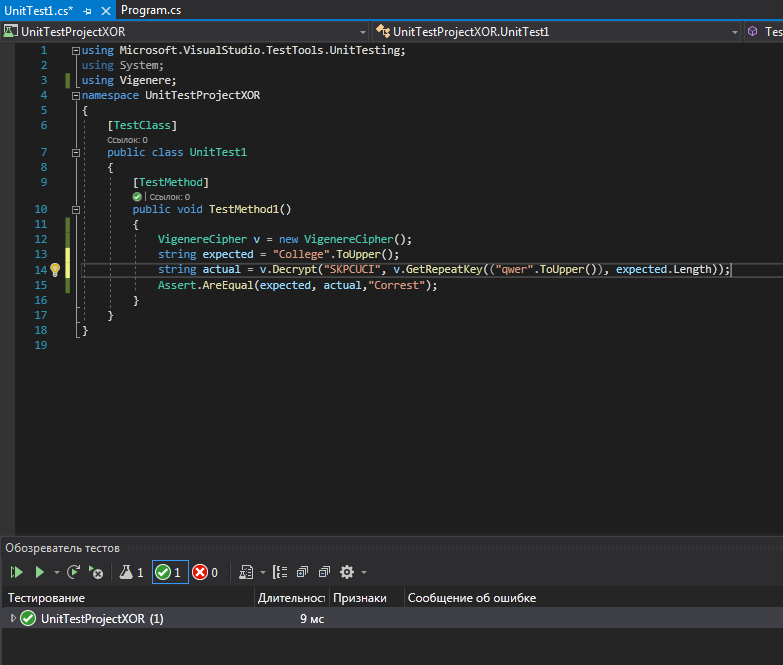


Рис.21. Скриншот из Visual Studio

1. Тестирование функций encrypt и decrypt, класс DES.

Работа функции заключается в шифровании строки по ключу и ее

расшифровке по алгоритму шифрования DES.

На Рис.22 мы видим, что тест пройден, функция decrypt вернула строку

"Бэтмэн" как и ожидалось. Функция работает корректно.

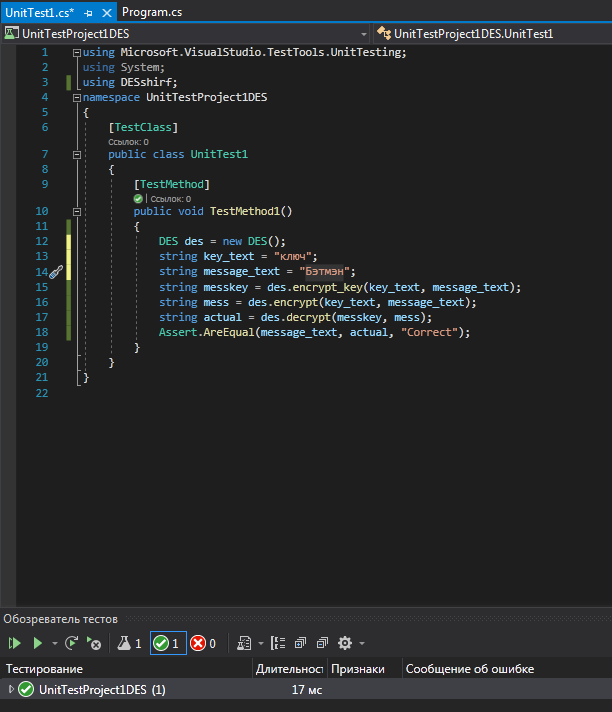


Рис.22. Скриншот из Visual Studio

На Рис.23 мы видим, что тест пройден, функция decrypt вернула строку

" Бэтмэн Betmen 123" как и ожидалось, что говорит о том, что функции могут шифровать и дешифровать буквы английского и русского алфавита вместе с цифрами. Функция работает корректно.

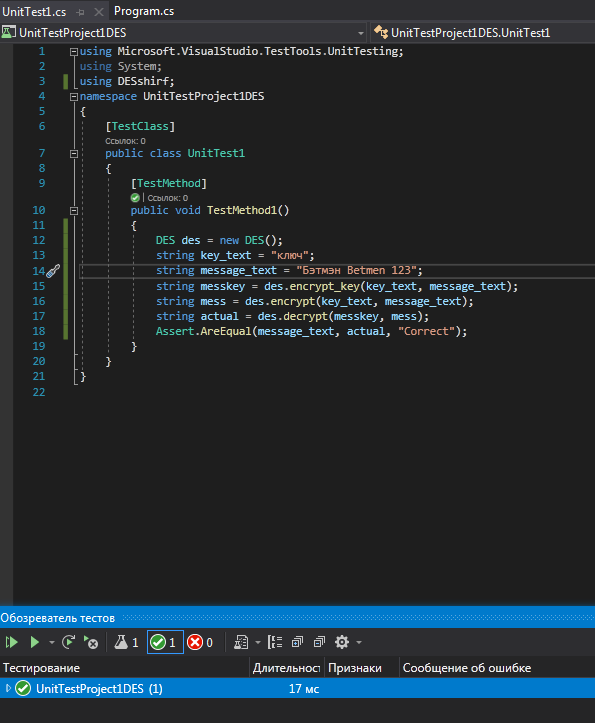


Рис.23. Скриншот из Visual Studio

**Вывод**: изучили методы подготовки и проведения тестирования и получили

навыки создания и выполнения тестов для приложений и их компонентов.

**Контрольные вопросы**:

1. Что такое тестирование, как его выполняют? Какие виды тестирования бывают?

Тестирование — процесс исследования, испытания [программного продукта](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определенным образом

Виды: Модульные тесты, Интеграционное тестирование, Функциональные тесты, Сквозные тесты, Приемочное тестирование, Тестирование производительности, Smoke-тестирование

1. Что такое верификация и валидация ПО?

Верификация — подтверждение на основе представления объективных свидетельств того, что установленные требования были выполнены.

Валидация — подтверждение на основе представления объективных свидетельств того, что требования, предназначенные для конкретного использования или применения, выполнены.

1. Что содержит тестовый вариант/сценарий ?

Тестовый сценарий определяется как любой функции , которые могут быть проверены. Это также называется условием проверки или возможностью проверки . Как тестер, вы должны поставить себя на место конечного пользователя и выяснить реальные сценарии и варианты использования тестируемого приложения.

1. Какие виды ошибок выявляют тесты?

Неправильная постановка задачи, неверный алгоритм, ошибка анализа, семантические ошибки, синтаксические ошибки, ошибки при выполнении операций, ошибки в данных, опечатки, ошибки ввода-вывода.

Отчёты об ошибках часто включают в себя такую информацию, как: тип крэша, образ [стека](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BA), версия программы, тип центрального процессора, версия [операционной системы](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), а также лог программы.

1. Что такое успешный тест и хороший тест?

Успешный тест - это тест который прошел по плану как нужно, хороший тест это тест, который написан правильно.

1. Что такое модульное тестирование? Его цели и выявляемые ошибки.

Модульное тестирование - это тестирование какой то части программы (модуля). Цель: изолировать отдельные части программы и показать, что по отдельности эти части работоспособны. Ошибки: тестирование больше одной вещи за раз, принятие в расчет внешних сервисов, цель — покрытие тестами.

1. Средства и методы проведения модульных тестов.

Модульное тестирование в TDD включает в себя широкое использование платформ тестирования. Каркас модульного тестирования используется для создания автоматизированных модульных тестов. Структуры модульного тестирования не являются уникальными для TDD, но они необходимы для него. Заявление покрытия, охват решений, охват филиала, состояние покрытия, покрытие конечного автомата.

1. Как определить покрытие кода? Зачем оно нужно?

Метод определения полноты системы тестов является определение отношения количества тест-требований, для которых существуют тестовые примеры, к

общему количеству тест-требований, Это полезный показатель позволяет оценить качество комплекта тестов.

1. Что такое нагрузочное тестирование? Его цели? Как можно его провести?

Нагрузочное тестирование — подвид тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе. Цель - установление соответствий требованиям, предъявляемым к системе. Можно провести просто использую максимальные ресурсы проекта.

1. Что такое тестирование пользовательского интерфейса? Его цели? Как его проводят?

Это тестирование позволяющее оценить насколько удобен интерфейс в приложении. Цель - выявить какие могут быть неудобства в интерфейсе. Проверяют читабельный ли шрифт например или находится ли кнопка в удобном для нас месте и понятны ли ее действия если посмотреть на ее название.

1. Что такое профайлинг? Его цели и результаты? Как можно его провести.

Профайлинг — инструмент психодиагностики, который позволяет составить комплексный психологический портрет человека, чтобы предсказать его поведение в той или иной ситуации. Цель - определить правдивость информации собеседника.

1. Зачем и как проводят автоматизацию тестирования? Что такое тестовый драйвер и заглушка?

Автоматизация тестирования позволяет значительно ускорить выпуск программного продукта и сократить стоимость тестовой итерации. В современном мире скорость выпуска программных продуктов на рынок является неотъемлемой частью конкурентной борьбы.

Проводятся с использованием специальных программных средств, которые, в свою очередь необходимы для выполнения набора тестовых примеров.

Драйвер - определенный модуль теста, который выполняют тестируемый нами элемент.

Заглушка — это небольшая часть кода, которая заменяет собой другой компонент во время тестирования.

1. Виды и типы тестирования? Методы тестирования?

Функциональные и нефункциональные.

Модульное тестирование (Unit testing)

Интеграционное тестирование (Integration testing)

Системное тестирование (System testing)

Приемочное тестирование (Acceptance testing)

Тестирование методом черного ящика (Black-box testing)

Тестирование методом белого ящика (White-box testing)