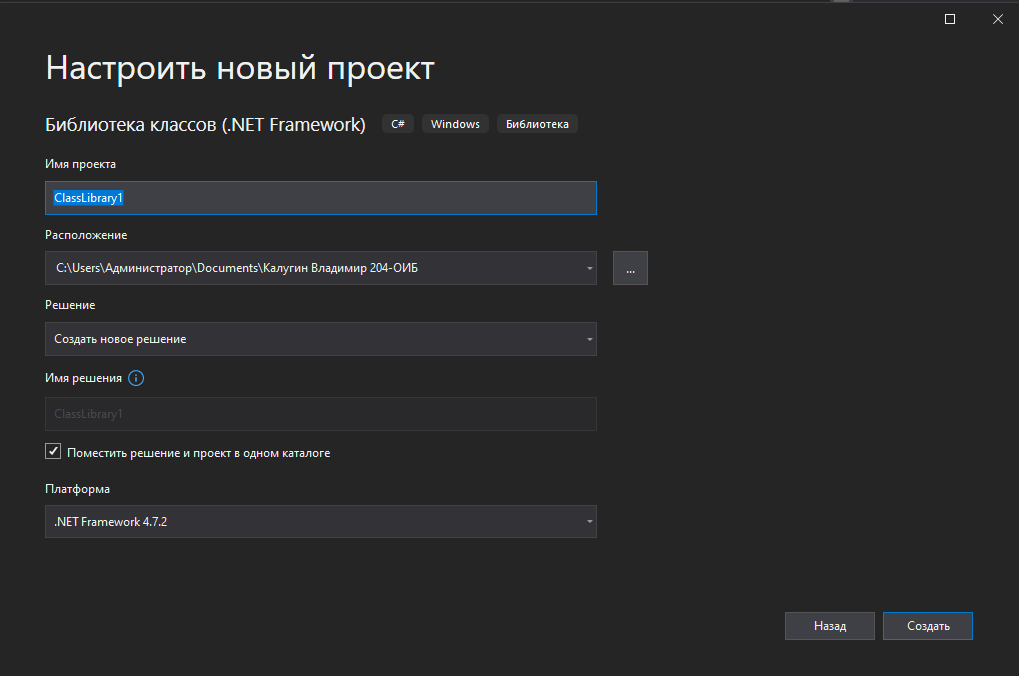
**Создание Unit test**

*Работу выполнил: Комаров Данила*

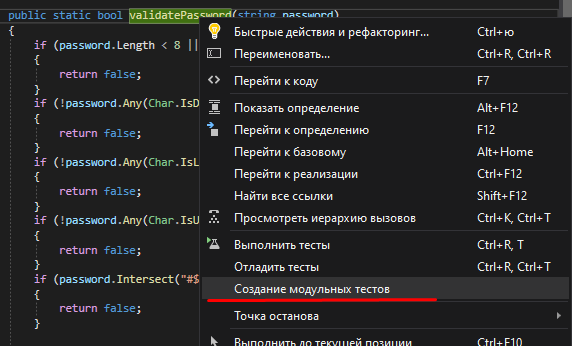
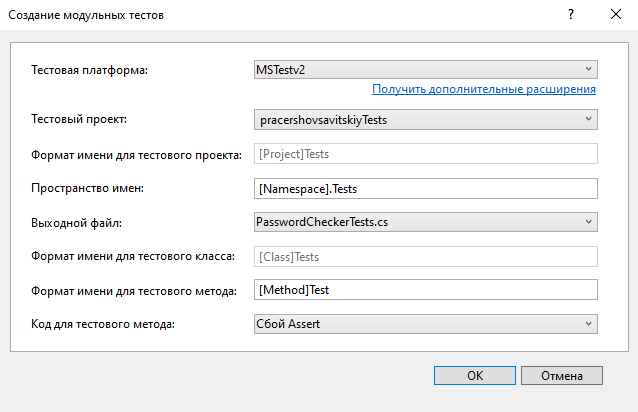
*Группа: 195 ИС и Программирование*

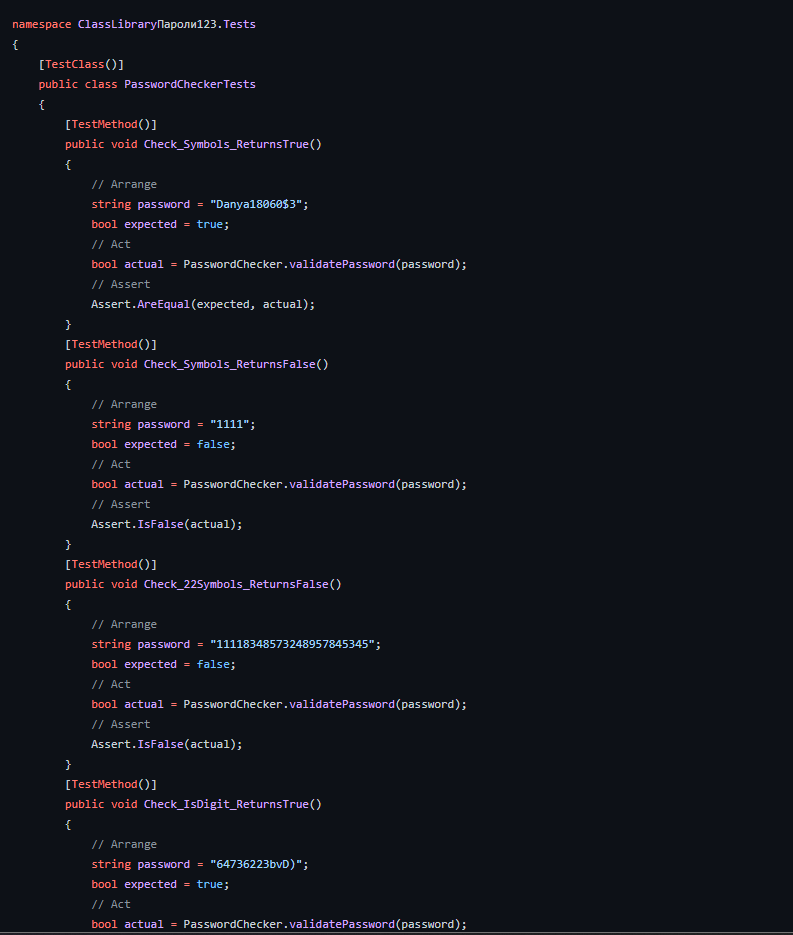
|  |  |  |
| --- | --- | --- |
| **Требование** | **Пароль** | **Результат** |
| Кол-во символов 8<>20 | Danya18060$3 | True |
| 1111 | False |
| 11118348573248957845345 | False |
| Наличие цифр | 64736223bvD | True |
| hds | False |
| Наличие спец. Символов(#$%@\_-\*&^) | PEIWODOa$2S | True |
| DANEK | False |
| Наличие прописных и строчных букв | dAneche18(k | True |
| kakdela | False |
| Все критерии | cAd1diLLAc&$ | TRUE |

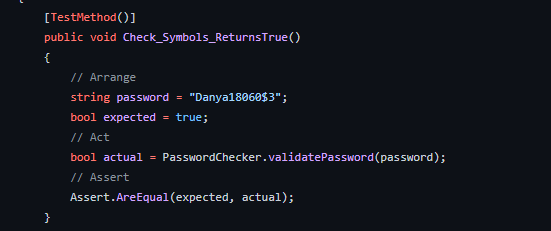
1. Цель:
   1. Научиться работать с методами
   2. Научиться проводить тестирование
   3. Научиться создавать условия проверки
2. Задачи:
   1. Создать программу проверку паролей в соответствии с заданными критериями:
3. *Количество символов не может быть меньше 8 и не может быть больше 20*
4. *В пароле должны присутствовать цифры*
5. *В пароле должны присутствовать спец. символы*
6. *В пароле должны быть прописные и строчные буквы*
   1. Провести тестирование
7. Описание:
   1. Была создана таблица
   2. Введены данные
   3. Создан проект VS C# «Библиотека классов C# (.Net)
   4. Создана функция validatePassword



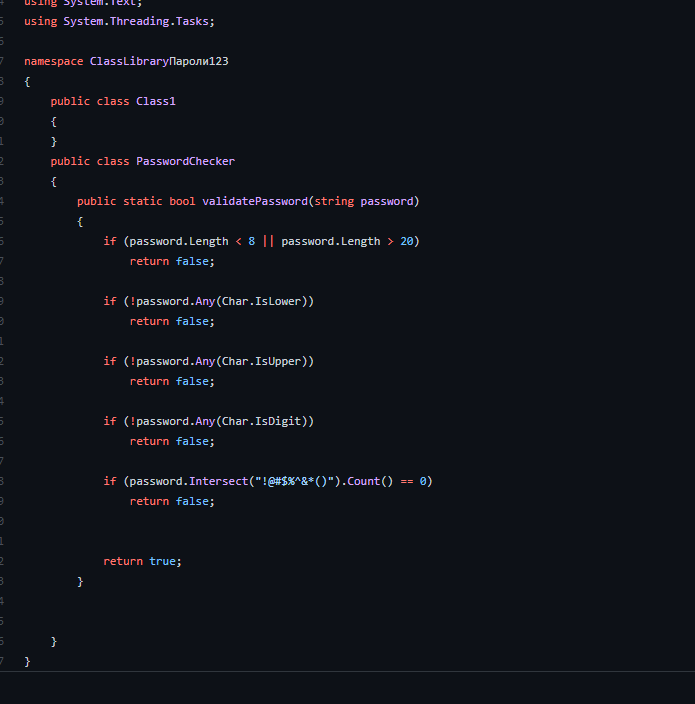
* 1. Создан Модульный тест нажатием пкм по имени функции -> создание модульных тестов

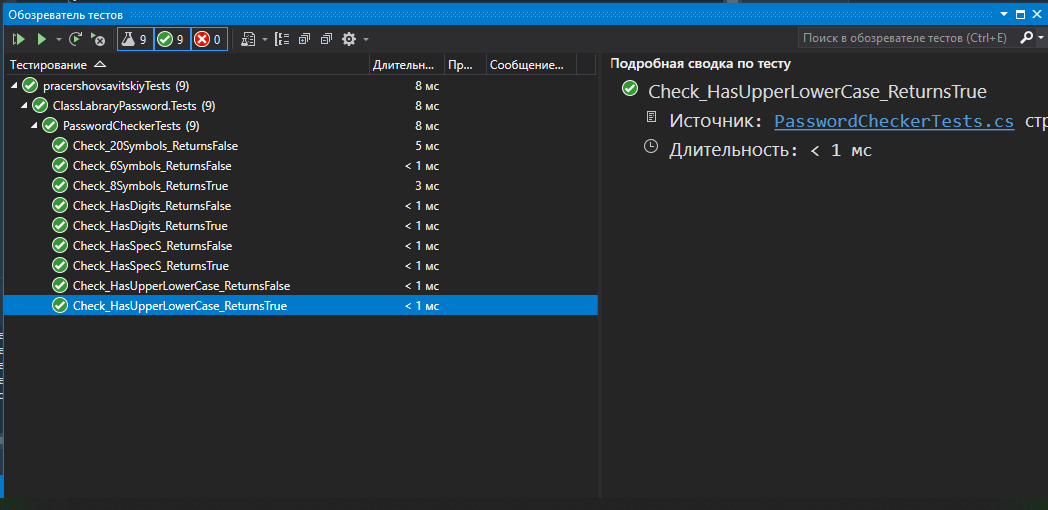
* 1. Добавлены методы проверки 
     1. В блоке Arrange объявлена переменная для пароля тестовых данных и ожидаемое значение в результате выполнения теста
     2. В блоке Act создаем переменную, возвращающую актуальный результат при выполнении метода PasswordChecker validatePassword
     3. И с помощью Assert сравниваем два значения: ожидаемое и реальное, Метод AreEqual, в кач-ве аргумента – наши данные.



* 1. Проведено тестирование
  2. Получены первые результаты по условию true
  3. Созданы функции условия



* 1. Проведено повторное тестирование
  2. Получены удовлетворительные результаты



* 1. Проверены результаты

1. Для выполнения задания следует знать следующие действия:
   1. Как работают булевы функции
      1. Такие операции возвращают логическое значение, то есть значение типа **bool**: **true**, если выражение истинно, и **false**, если выражение ложно. К подобным операциям относятся операции сравнения и логические операции:

== Сравнивает два операнда на равенство

!= Сравнивает два операнда и возвращает true, если операнды не равны, и false, если они равны.

< Операция "Меньше чем".

> Операция "Больше чем".

<= Операция "меньше или равно"

>= Операция "Больше или равно"

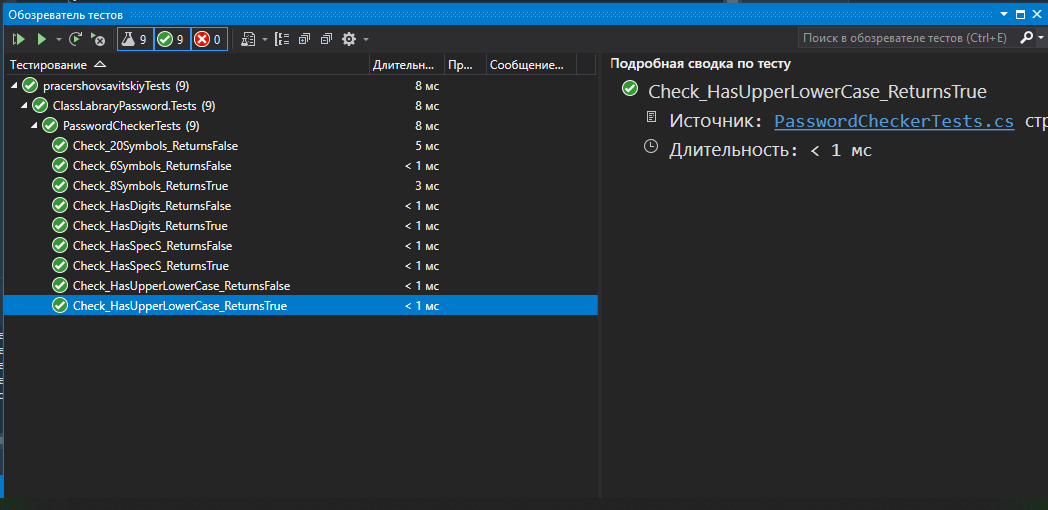
* 1. Как работают методы проверки условия
  2. Знать о таких методах, как:
     1. Метод **Char.IsDigit**: показывает, относится ли указанный символ Юникода в указанной позиции в указанной строке к категории десятичных цифр.
     2. Метод **Char.IsLower**: показывает, относится ли символ Юникода к категории букв нижнего регистра.
     3. Метод **Char.IsUpper**: показывает, относится ли символ Юникода к категории букв верхнего регистра.
     4. Метод **Intersect** - Пересечения последовательностей:

Позволяет из двух последовательностей объектов сгенерировать набор уникальных элементов.

* + 1. Метод **Any**: Методы *All*, *Any* и *Contains* позволяют определить, соответствует ли коллекция определенному условию, и в зависимости от результата они возвращают true или false.
       1. Метод **All** проверяет, соответствуют ли все элементы условию
       2. Метод **Any** действует подобным образом, только позволяет узнать, соответствует ли хотя бы один элемент коллекции определенному условию

1. Результаты:

В результате программа проводит проверку пароля по всем критериям, указанным в обозревателе: кол-во символов, наличие цифр, спецсимволов, строчных и прописных букв.



Результат: [-195-/1ЮнитТест/класспароль at main · DanekTerminator/-195- (github.com)](https://github.com/DanekTerminator/-195-/tree/main/1%D0%AE%D0%BD%D0%B8%D1%82%D0%A2%D0%B5%D1%81%D1%82/%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BF%D0%B0%D1%80%D0%BE%D0%BB%D1%8C)