**Лабораторная работа №8**

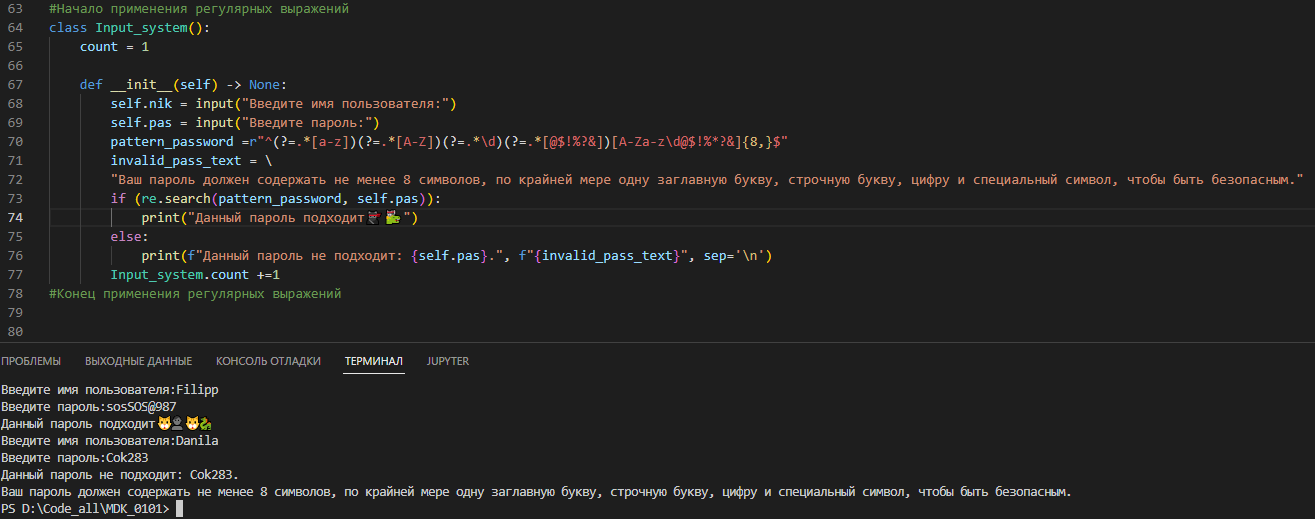
**ИСПОЛЬЗОВАНИЕ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ. ОПЕРАЦИИ СО**

**СПИСКАМИ**

**Цель работы**

Научиться использовать регулярные выражения.

**Ход работы**



**Рисунок 1 –** реализация класс входа в систему с проверкой пароля

**Контрольные вопросы**

1. **Основные методы для работы с регулярными выражениями**

* Методы класса Regex позволяют выполнять следующие операции:

Определить, входит ли шаблон регулярного выражения во входной текст, с помощью метода Regex.IsMatch. Пример использования метода IsMatch для проверки текста см. в разделе Руководство. Проверка строк на соответствие формату электронной почты.

* Получить один или все экземпляры текста, соответствующего шаблону регулярного выражения с помощью метода Regex.Match или Regex.Matches. Первый метод возвращает объект System.Text.RegularExpressions.Match, который предоставляет сведения о соответствующем тексте. Второй метод возвращает объект MatchCollection, содержащий один объект System.Text.RegularExpressions.Match для каждого соответствия, обнаруженного в обработанном тексте.
* Заменить текст, соответствующий шаблону регулярного выражения, с помощью метода Regex.Replace. Примеры использования метода Replace для изменения форматов даты и удаления недопустимых символов из строки см. в разделах Руководство. Исключение недопустимых символов из строки и Руководство. Изменение форматов даты.

2. **Экранирование спец символов.**

|  |  |
| --- | --- |
| \newline | Если после символа "\" сразу нажать клавишу *Enter* то это позволит продолжать запись с новой строки. |
| \\ | Позволяет записать символ обратного слеша. |
| \' | Позволяет записать один символ апострофа. |
| \" | Позволяет записать один символ кавычки. |
| \a | Гудок встроенного в систему динамика. |
| \b | Backspace, он же возврат, он же "пробел назад" – удаляет один символ перед курсором. |
| \f | Разрыв страницы. |
| \n | Перенос строки (новая строка). |
| \r | Возврат курсора в начало строки. |
| \t | Горизонтальный отступ слева от начала строки (горизонтальная табуляция). |
| \v | Вертикальный отступ сверху (вертикальная табуляция). |
| \xhh | Шестнадцатеричный код символа (две шестнадцатеричные цифры *hh*). |
| \ooo | Восьмеричный код символа (три восьмеричные цифры *ooo*). |
| \0 | Символ *Null*. |
| \N{id} | ID (идентификатор) символа в базе данных Юникода, или, проще говоря, его название в таблице Юникода. |
| \uhhhh | Шестнадцатеричный код 16-битного символа Юникода (символ кодируемый двумя байтами). |
| \Uhhhhhhhh | Шестнадцатеричный код 32-битного символа Юникода (символ кодируемый четырьмя байтами). |
| \*other* | Под *other* понимается любая другая последовательность символов. Не является экранированной последовательностью (остается без изменений с сохранением в строке символа "\"). |

**Вывод**

Я научился использовать регулярные выражения.