**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая Кибернетика и Информационные технологии»

Дисциплина «Информационные технологии и программирование»

Лабораторная работа №5

«Регулярные выражения в Java»

Выполнила:

Студентка группы БВТ2303

Морозова Ольга

**Цель работы:**

Изучение регулярных выражений и применение полученных знаний на практике на языке программирования Java.

**Ход работы:**

Задание 1.

Напишем регулярное выражение для поиска всех чисел в заданном тексте (положительных и отрицательных целых и дробных (натуральных и десятичных) чисел).

Используем экранированный метасимвол \d для поиска числовых символов.

Используем [./] для указания, что встречается один из перечисленных в скобках символов.

Также используем квантифакторы, обозначающие количество символов:

\* – ноль или более раз;

+ – один или более раз;

? – один раз или отсутствует;



Задание 2.

Напишем регулярное выражение для проверки корректности введённого пароля. Пароль должен состоять из латинских букв и цифр, быть длиной от 8 до 16 символов и содержать хотя бы одну заглавную букву и одну цифру.

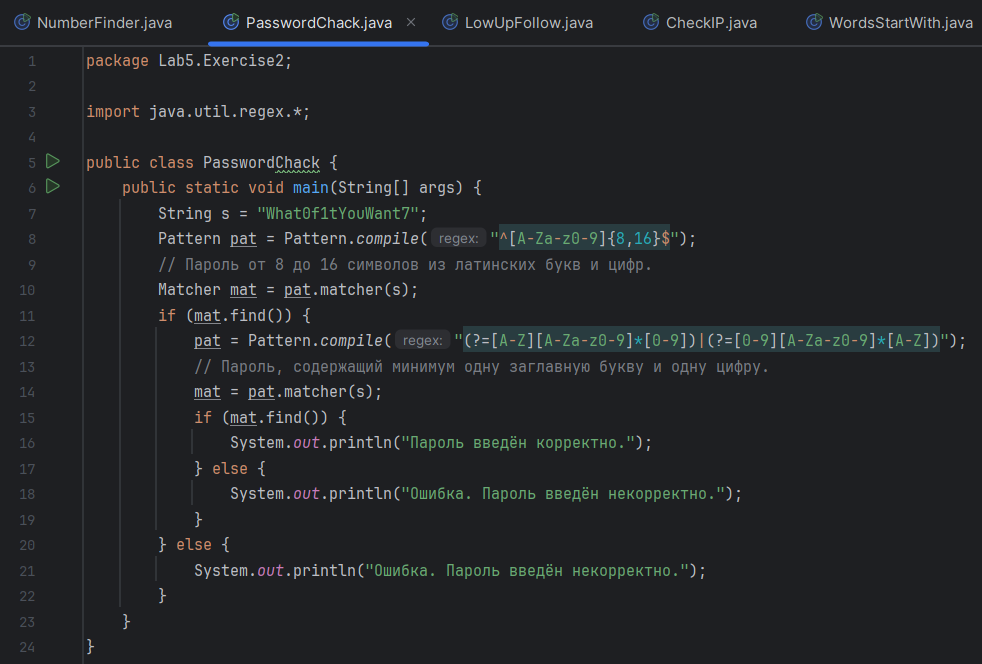
Первое регулярное выражение проверяет количество символов в пароле и их содержание (только латинские буквы (заглавные и строчные) и цифры).

Для этого используем квантифактор {8,16} – значит повторение символа (одного из указанных в квадратных скобках) будет от 8 до 16 раз.

Можно указывать просто {8} – это точное число повторений, или же {8,} – от 8 раз и более, например.

И обязательно при проверке длины указываем символы начала «^» и конца «$» строки.

Второе регулярное выражение проверяет чес помощью логического ИЛИ - ()|() – обязательное наличие одной заглавной буквы и одного числа (ИЛИ мы используем, чтобы указать два варианта – когда заглавная буква стоит в пароле до числа или после него).

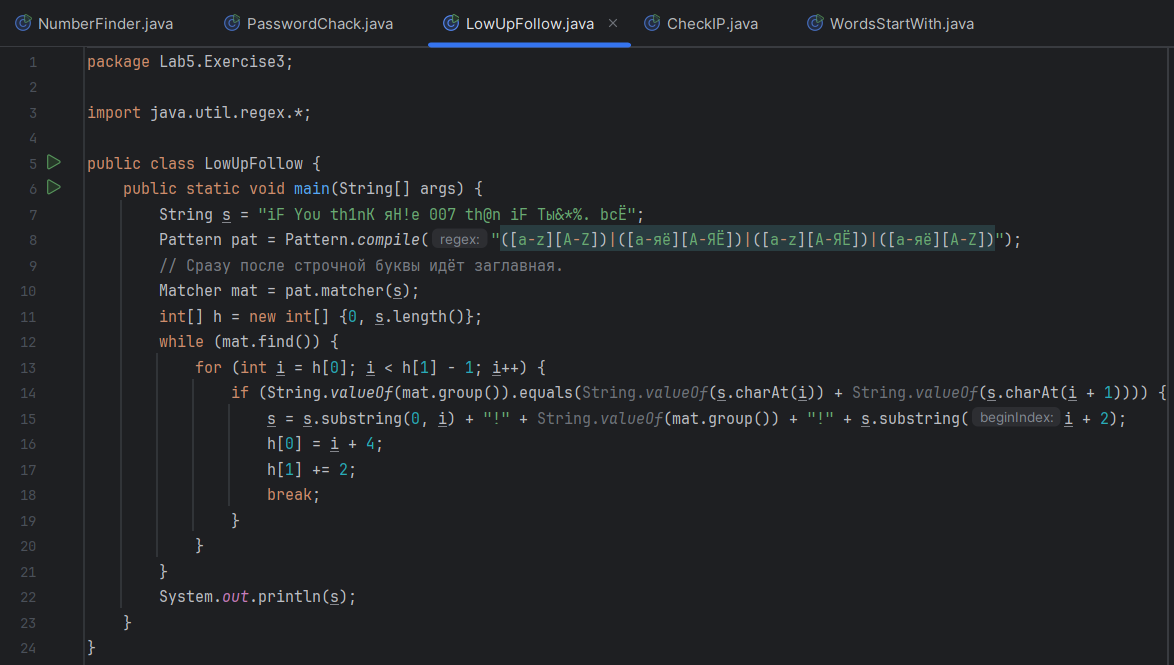


Задание 3.

Напишем регулярное выражение для нахождения всех случаем в тексте, когда сразу после строчной буквы идёт заглавная а после программой выделим эти случае в тексте знаками «!» с двух сторон.

К слову, в диапазон русских букв от А до Я (что строчных, что заглавных) не включена буква Ё, поэтому пришлось прописывать её дополнительно в группах.

Замену производим с помощью двух циклов (while и for) для прохождения по всем найденным случаям и после нахождения их в строке с последующим преобразованием оной.

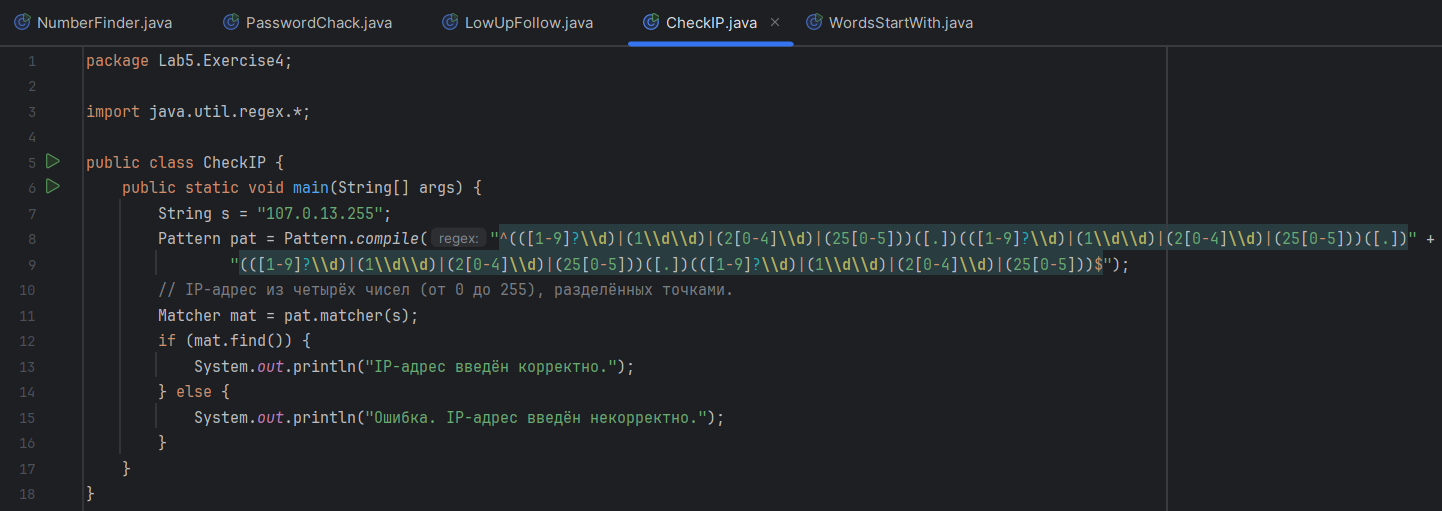


Задание 4.

Напишем регулярное выражение для проверки корректности введённого IP-адреса. IP-адрес должен состоять из 4 чисел, разделенных точками, и каждое число должно быть в диапазоне от 0 до 255.

Обязательно при проверке длины указываем символы начала «^» и конца «$» строки.

Через ИЛИ проверяем все варианты для каждой из 4 групп чисел (чтобы исключить ведущие нули и не превысить порог в 255). Между ними ставим обязательные точки.



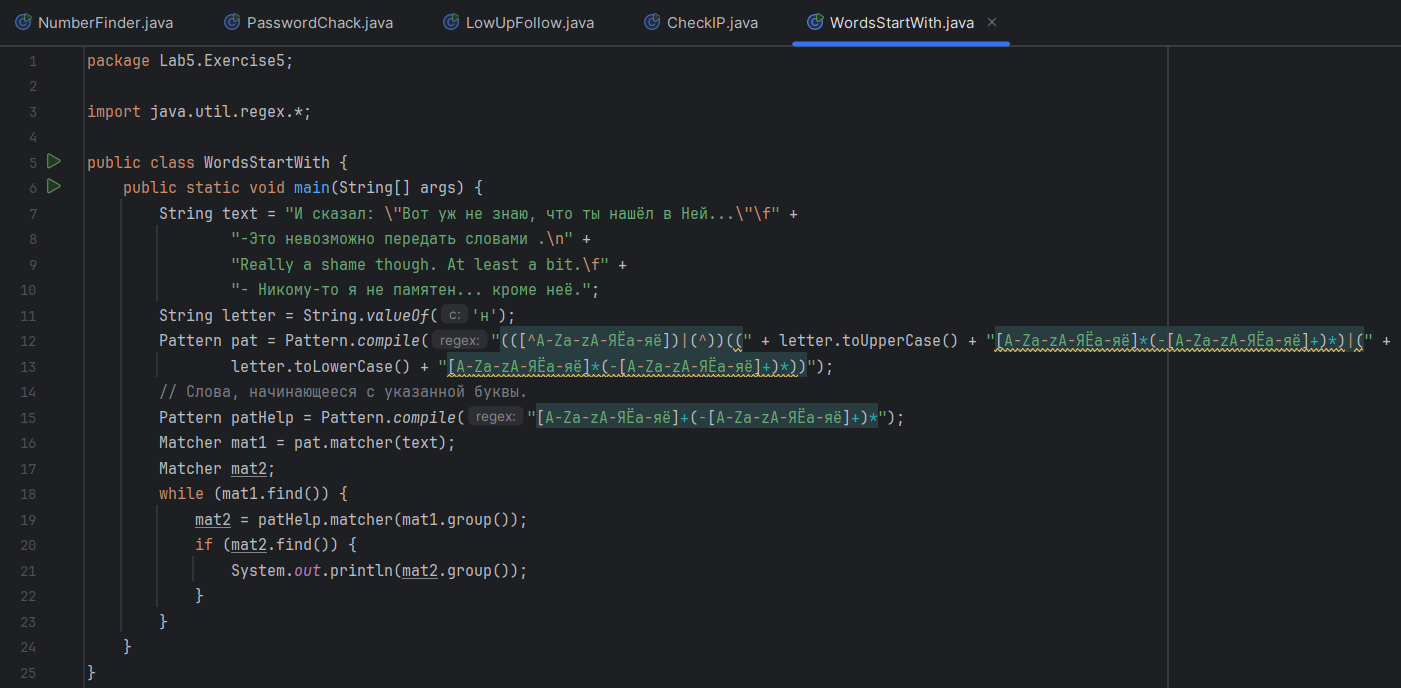
Задание 4.

Напишем регулярное выражение для поиска всех слов в заданном тексте, начинающихся с заданной буквы.

Используем конкатенацию строк, чтобы вставить значение переменной в регулярное выражение. Причём с помощью ИЛИ мы делаем одну и ту же проверку для заданной буквы сперва в её заглавном виде, а потом в строчном.

Чтобы определить начало слова используем [^…] с указанием внутри всех латинских и русских букв (заглавных и строчных), тем самым указывая, что символ перед желаемой первой буквой слова не должен являться буквой. Или же это должно быть начало строки.

Вторым же регулярным выражением мы просто избавляемся от этого первого “небуквенного” символа, чтобы на выходе получить чистые слова.



**Вывод:**

Мы изучили регулярные выражения, а также написали программы на языке программирования Java с использованием полученных знаний.

GitHub - https://github.com/MiniLynx13/ITaP\_Lab5