Вы можете сделать регулярное выражение, в котором какие-то знаки могут присутствовать а могут и отстуствовать, совпадение в любом случае будет найдено. Для этого после знака ставится знак вопроса: **?**. И если мы хотим находить в тексте номера и с дефисами и без, наше регулярное выражение примет вид:

**8029-?222-?24-?60**

Такое выражение будет соответствовать и номеру **8029-222-24-60** и номеру **8029-222-2460** и номеру **80292222460**.

Таким образом, некоторые знаки в регулярных выражениях обозначают не буквы, а операции, такие как количество возможных повторений знака и т.д. Такие символы называются метасимволами. Метасимволами в РВ являются следующие знаки: **^ $ \ . \* + ? ( ) [ ] { } |**

Чтобы превратить знак вопроса из метасимвола в обычную букву, следует поставить перед ним знак косой черты (слеш) **\**. И, например, если вам необходимо найти в тексте фрагменты с тремя идущими подряд знаками вопроса, регулярное выражение будет иметь вид:

**\?\?\?**

Чтобы любой метасимвол воспринимался просто как буква, перед ним надо ставить слеш. Это часто называют экранированием.

Требование ставить слеш касается и его самого. Так если вам необходимо найти в тексте строку вида: **c:\folder\file.txt**, регулярное выражение для этого будет иметь вид:  
**c:\\folder\\file.txt**.

### Классы символов

Например, нам необходимо найти двузначное число в тексте, у которого первый знак — единица, а второй — любая цифра. Регулярное выражение будет иметь вид:

**1[1234567890]**

Также, как после обычного символа, после квадратных скобок можно ставить знак вопроса, который означает, что здесь может не быть ни одного символа.

Сокращенные перечисления:

**[0-9]**

**[a-z]** идут подряд в текущей кодировке

**[a-zA-Z]** абсолютно любой буквы латинского алфавита

**[а-яА-Я]**

Но есть одна буква, использование которой требует дополнительных усилий. Это буква **Ё**. Она в кодировках расположена отдельно от основного алфавита, и тогда выражение для любой буквы русского алфавита будет иметь вид: **[а-яА-ЯёЁ]**. Это же касается особых букв других кирилличных алфавитов, в частности белорусских **Ў**,**і**. Обращаем внимание — последняя буква и латинское **i** в большинстве кодировок являются разными символами.

id="[^"]\*" - все символы кроме “

<p id="test">Тест</p><div id="p"><div><img src="im1" id="m2">

чтобы задать цифру, можно написать:

**\d** равнозначна **[0-9]**.

Если вам надо указать, что в данном месте должен стоять пробел, но кроме пробела там может быть и табуляция и перевод строки и другие пробельные символы, можно написать:

**\s**

Существует также класс для буквы:

**\w**

Но следует помнить, что под буквой здесь понимаются не обычные буквы. **\w** равнозначен классу **[a-zA-Z\_0-9]**. То есть под буквами здесь понимаются буквы латинского алфавита, цифры, и знак подчеркивания. Для кириллицы это не работает!

Для каждого из этих классов есть обратный, он записыватся также, но буква пишется в верхнем регистре. **\D** означает "любой символ кроме цифры", **\S** — любой непробельный символ и т.д.

Иногда необходимо в регулярном выражении указать, что здесь может стоять любой символ. Для этого используется символ точки **.**.

\* - любое количество

**#.\***

В данном регулярном выражении ищется знак решетки, а затем любое количество любых знаков (цифры, буквы, пробел \_) до конца.

Мы уже знакомы с вами с одним из символов повторений: это знак вопроса??????. Если его поставить после символа, это будет означать, что данный символ будет присутствовать либо один раз либо ни одного.

И если у нас задача найти числа, которые могут содержать от 1 до 5 знаков, мы можем записать следующее выражение:

**\d?\d?\d?\d?\d**

В этом выражении, после каждого из знаков, кроме последнего, стоит знак вопороса, значит, они могут отсутствовать, и таким образом будет найдено любое число, длиной от 5 до 1 цифры.

Плюс, стоящий после символа означает, что символ может присуствовать один и больше раз, верхнее количествон не ограничено. И предыдущее регулярное выражение записывается следующим образом:

**\d+**

Такое выражение найдет все целые числа в тексте.

Бывают случаи, когда символов может быть много, но может не быть ни одного. В этом случае после повторяющегося символа ставится звездочка **\***

Например, вам необходимо найти слово **стол**, но не просто это слово, но и все падежи этого слова. Самым простым вариантом может быть следующее регулярное выражение:

**стол[а-я]\***

Такое регулярное выражение найдет и **стол** и **столу** и **столом** и т.д.

Чтобы задать точное количество повторений, после символа или класса надо поставить фигурные скобки с числом повторений в них.

**[0-9]{3}**

Данная запись означает, что здесь может стоять три любых цифры.

квадратные скобки, в которых через запятую указывается максимальное и минимальное количество повторений:

**[0-9]{2,4}**

Такое регулярное выражение найдет числа **15** **129** **3456**, но не найдет **5**

если необходимо, количество повторений от какого-то числа и до бесконечности, в фигурных скобках ставится число и запятая, второе число не указывается

**[А-Яа-я]{3,}**

Для того, чтобы создать группу, фрагмент регулярного выражения следует взять в круглые скобки. Например, вы ищите в тексте дробные числа, и вам затем понадобится отдельно работать с целой и дробной частями. В этом случае, регулярное выражение примет вид:

**(\d+)\.(\d+)**

В этом регулярном выражении сначала находится одна и более цифр, затем знак точки, затем еще одна и более цифр. Такое выражение найдет числа вида: **1.5** **387.77** и т.д.

Все группы нумеруются начиная с 1.

Обратите внимание, что круглые скобки являются метасимволами, и записанные просто так они не ищутся. Т.е. текст вида **(1).(12)** данное регулярное выражение не найдет. Чтобы искать знаки круглых скобок, их необходимо экранировать слешами: **\(\)**

**\d+\.\d+**

Это регулярное выражение будет находить в тексте те же самые фрагменты, что и предыдущее, но использовать результат его работы можно будет только целиком.

Результаты группировки можно использовать в самом регулярном выражении. Если у вас в начале есть группа, и в конце должен повториться текст, найденный в этой группе, вы можете воспользоваться ссылкой. Ссылка пишется как слеш, после которого ставится номер группы например, ссылка на первую группу в выражении: **\1**. Рассмотрим пример: нам необходимо найти в тексте числа начинающиеся и заканчивающиеся одной и той же цифрой. Регулярное выражение будет иметь вид:

**(\d)\d\*\1**

Это выражение сначала ищет цифру, помещая ее в группу, затем любое количество других цифр, и в конце содержимое первой группы, то есть цифру, стоявшую на первом месте. Такое выражение найдет текст **22** **151** **287892**, но не найдет **115** **456**.

Еще один пример: необходимо найти симметричные четырех и пятизначные числа в тексте. Регулярное выражение будет иметь вид:

**(\d)(\d)\d?\2\1**

Такое выражение найдет числа **12321** **2222**, но не найдет **22** **12341**.

гуппировка с выбором. Выглядит это следующим образом: в круглых скобках помещается несколько вариантов текста с разделителем в виде вертикальной черты **|**.

И регулярное выражение решающее такую задачу будет иметь вид:

**80(29|17)[ -][0-9]{3}[ -][0-9]{2}[ -][0-9]{2}**

Такое выражение будет соответствовать и номеру **8029-222-24-60** и номеру **8017-222-24-60** но не номеру **8033-222-24-60**.

**(ля){3}**

Такое регулярное выражение найдет в текст **ляляля**.

Если необходимо сделать привязку к началу строки, в начало строки ставится знак **^**. Например, вам необходимо найти слово с которого начинается кирилличный текст. Регулярное выражение для этого будет иметь вид:

**^[а-яА-Я]+**

То есть ищется последовательность идущих подряд кирилличных букв, но только в начале строки. И если у вас исходная строка **Только этот текст**, данное регулярное выражение найдет слово **Только**, и не найдет остальные.

Обратите внимание, что знак **^** имеет разное значение, когда находится просто в тексте регулярного выражения, и в квадратных скобках. В первом случае это привязка к началу, во втором случае — отрицание. Причем, в обоих случаях он должен стоять первым.

Если вам нужен просто знак **^**, его надо использовать с косой чертой:**\^**.

Если необходимо сделать привязку к концу строки, в конце регулярного выражения следует поставить символ **$**. В этом случае, будет найдено совпадение с регулярным выражением, только если оно завершает текст. Например, нам необходимо найти слово в конце строки (учтем, также, что после слова может стоять знак точки, запятой или пробельный):

**[а-яА-Я]+[\.,\s]?$**

Это выражение сначала ищет последовательность кирилличных букв, после которой может следовать запятая, точка или пробел, и такая последовательность будет найдена исключительно в конце строки.

Можно совместить привязку к концу строки и к началу. Достаточно часто встречаются задачи, когда надо проверить соответствует ли данному регулярному выражению строка целиком. В этом случае, в начале выражения ставится **^** а в конце **$**. Например, нам необходимо убедиться, что в строке нет ничего кроме цифр. регулярное выражение для этого будет иметь вид:

**^[0-9]+$**

При использовании такого выражения будет найдено соответствие, если в строке только цифры. Если есть хотя бы один другой знак, например **1234r4321**, соответствия найдено не будет.

Не следует использовать неэкранированные знаки **^** **$** внутри регулярного выражения. Это считается ошибкой, и результат будет зависеть от программы, или библиотеки, с помощью которой выполняется работа с РВ.

Регулярные выражения можно привязывать к словам. Для этого в начало ставится спецсимвол **\b**. В этом случае совпадение должно начинаться с начала слова. Например, вам необходимо найти в тексте слово **compression**, но, чтобы при этом не был найден фрагмент слова **decompression**. Регулярное выражение будет иметь вид:

**\bcompression**

Это выражение находит текст **compression** перед которым стоят любые символы кроме символов латинского алфавита, знака подчеркивания и цифр. Т.е. чтобы сработал **\b** перед словом должно стоять что угодно, кроме символов **\w**.

Следует помнить, что в регулярных выражениях символы кириллицы и другие национальные символы буквами не считаются, поэтому в тексте **проверкаcompression** слово **compression** будет найдено. То есть для русских букв привязка к словам не работает.

Привязку к словам можно ставить как в начале так и в конце регулярного выражения. То есть, если при анализе текста вас интересуют только отдельно стоящие слова, следует ставить **\b** и в начале и в конце.

При этом цифры и знак подчеркивания считаются буквами, поэтому если у вас стоит задача найти отдельно стоящие числа в английском тексте, можно воспользоваться выражением вида:

**\b[0-9]+\b**

Такое регулярное выражение найдет отдельно стоящие числа **12345 5**, но не найдет числа записаные вместе с латинскими буквами: **number12**.

Как и у других специальных символов, у символа привязки к словам есть обратная запись: **\B**. Если этот спецсимвол стоит в начале регулярного выражения, это означает, что совпадение не должно начинаться с начла слова. Например, в тексте имеются некоторое количество чисел записаных в виде: **n15**, **n3** и т.д. Нам необходимо найти сами числа, при этом мы не должны находить обычные отдельностоящие числа. Здесь нам пригодится именно такая привязка:

**\B[0-9]+**

Такое регулярное выражение найдет числа склеенные с латинскими буквами но не найдет стоящих отдельно.

И точно также как с символом **\b**, следует помнить: символ **\B** не работает корректно с кириллицей и другими национальными алфавитами.

Когда выполняется поиск с помощью регулярных выражений, программы и библиотеки используют по-умолчанию "жадные" алгоритмы. Это означает, что совпадение должно быть самым большим из всех возможных. Допустим, у вас есть регулярное выражение вида:

**<.\*>**

И текст, в котором выполняется поиск выглядит следующим образом

**<b>Проверка?</b> жадного алгоритма**

В этом тексте регулярному выражению соответствуют и **<b>** и **</b>**, но найден будет текст:**<b>Проверка?</b>**. Будет захвачен наибольший фрагмент как единое целое. Т.к. поиск будет выполнен от самого первого знака **<** и до самого последнего знака **>**

Что делать, если вам на самом деле необходимо найти отдельно **<b>** и отдельно **</b>**?

Это делается с помощью уже знакомого нам метасимвола знака вопроса: **?**. Если он ставится после **\*** или **+**, значит алгоритм будет работать не в жадном режиме, и будет находить небольшие соответствия, а не искать наибольшее возможное.

**<.\*?>**

Когда после метасимвола плюс или звездочка ставится знак вопроса, включается режим который называют **ленивым** — **lazy**, при котором захватывается наименьшее необходимое количество знаков, чтобы текст соответствовал регулярному выражению.

Иногда требуется находить регулярное выражение только если оно стоит ПЕРЕД определенным текстом. Например, наша задача найти последние слова в предложениях. Это значит, необходимо найти идущие подряд буквы, после которых стоит один из знаков препинания: **.?!**. Но при этом сам знак препинания в найденную строку попадать не должен. Чтобы найти текст, стоящий перед каким-то другим следует использовать запись вида: **выражение(?=выражение2)**. В этом случае выражение будет найдено, только если после него стоит выражение2. Для нашей задачи с последними словами, регулярное выражение будет выглядеть так:

**[а-яА-Я]+(?=[.?!])**

Это выражение сначала ищет последовательность кирилличных букв, после которой обязательно стоит знак точка, или вопросительный, или восклицательный.

**\b** (как мы помним **\b** не работает для кирилличных текстов.)

В этом случае будет иметь вид:

**[а-яА-Я]+(?=[^а-яА-Я]|$)**

Обратите внимание, что знак **$**, хоть и не стоит на самой последней позиции регулярного выражения, в данном случае означает именно конец строки. Т.е. смысл этого РЕ можно сформулировать так: одна или больше букв кириллицы, за которой следует либо конец строки, либо любой знак, кроме букв кириллицы.

Существует и обратное условие, которое записывается в виде:**выражение(?!выражение2)**. В этом случае выражение будет найдено, только если за ним не следует выражение2. Например в тексте имеются двузначные номера аудиторий, и нас интересуют только номера без букв в конце. То есть нам надо найти **10** **15**, но не **10а** или **15г**. Регулярное выражение будет иметь вид

**[0-9]{2}(?![а-я])**

Существует также возможность сделать привязку не после, а до искомого текста. В этом случае выражение будет иметь вид **(?<=выражение2)выражение**. В этом случае выражение будет найдено, только если перед ним стоит выражение2.

Дополнительная инфа

<https://javarush.ru/groups/posts/regulyarnye-vyrazheniya-v-java>

<https://www.youtube.com/watch?v=_pLpx6btq6U>

