**Лабораторная работа 1**

Создайте проект. Задайте в нем следующую структуру пакетов (рис.1).

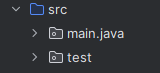


Рисунок 1. - Начальная структура проекта

В пакете main.java создайте класс Main. Он должен содержать функцию метод main (точку входа в проект). Здесь должна быть интерфейсная часть проекта. Туда же добавьте новый пакет с именем extensions. В нем создайте класс StringExtention. В данном классе будут реализованы методы из задач, описанных ниже.

В пакете test создайте класс StringExtentionTest. В этом классе будут написаны модульные тесты, доказывающие работоспособность методов класса StringExtention.

В результате проект должен выглядеть следующим образом (рис. 2).

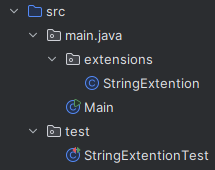


Рисунок 2. - Структура проекта

Далее нужно добавить библиотеку для работы с модульными тестами. Возьмем самую последнюю версию библиотеки JUnit5. Если предыдущие версии библиотеки (JUnit4 и ниже) можно было подключить одним jar-файлом, то здесь все сложно. Облегчим себе жизнь - доверимся IDE (она точно знает как надо сделать). С нашей стороны нужно просто выполнить следующие действия. Выберите меню File, далее Project Structure… В открывшемся диалоговом окне (рис. 3) выберите Libraries и добавьте библиотеку (кнопка +).

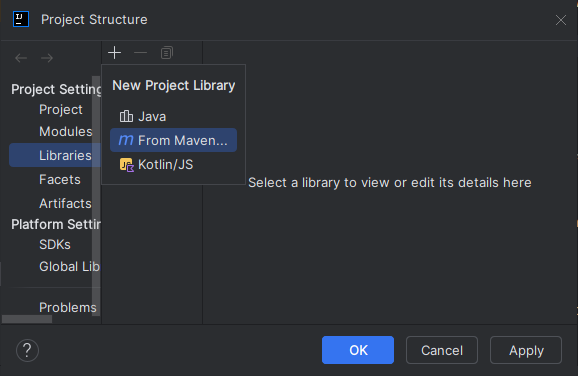


Рисунок 3. - Добавление библиотеки к проекту

Если у вас есть готовый jar-файл, скачанный на компьютер, то его можно добавить с помощью пункта Java. В данном случае будет использоваться From Maven…

В строке поиска введите название библиотеки. Данная библиотека на сайте mvnrepository.com известна как JUnit Jupiter (рис. 4). Возьмем версию 5.11.0.

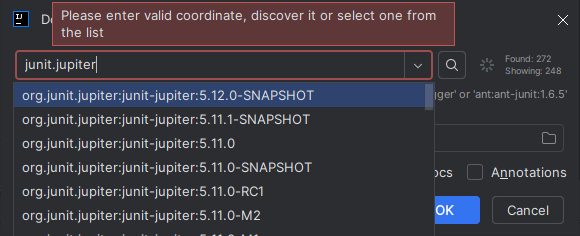


Рисунок 4. - Поиск библиотеки

Библиотека будет добавлена.

Если Вам все еще не понятно, что делать перейдите по ссылке в официальную документацию <https://www.jetbrains.com/help/idea/junit.html#intellij>. Перейдите по ссылке и ознакомьтесь!

Перейдем к написанию тестов. Подробнее о правилах работы с выбранной библиотекой можно прочитать по ссылке <https://junit.org/junit5/docs/current/user-guide/#overview-getting-started-features>.

Для старта достаточно знать, что тестовые методы в тестовом классе отмечены аннотацией @Test. Внутри тестового метода нужно использовать методы assertEquals, assertTrue и т.п.

Замечания.

1. Помните, что метод не должен ничего выводить, если это не требуется в задаче.
2. Предусмотрите достаточное количество примеров для полного покрытия кода тестами.
3. Интерфейсная часть (взаимодействие с пользователем) должна быть написана.

**Задания для самостоятельного выполнения**

1. Реализуйте метод transliteration принимающий в качестве аргумента исходную строку и строку-разделитель (divider по умолчанию " ") и возвращающий преобразованную строку из латинских символов. Создайте словарь символов соответствия алфавитов: "а": "a", "б": "b", "в": "v", "г": "g", "д": "d", "е": "e", "ё": "e", "ж": "zh", "з": "z", "и": "i", "й": "i", "к": "k", "л": "l", "м": "m", "н": "n", "о": "o", "п": "p", "р": "r", "с": "s", "т": "t", "у": "u", "ф": "f", "х": "h", "ц": "c", "ч": "ch", "ш": "sh", "щ": "sh'", "ъ": "", "ы": "i", "ь": "", "э": "e", "ю": "yu", "я": "ya".

Пример:

transliteration("Иван Иванов") //Ivan Ivanov

transliteration("Иванов John","\_") //Ivanov\_John.

1. Реализуй метод truncate, усекающий исходную строку до указанного числа символов (по умолчанию 16) и возвращающий усеченную строку с заполнителем "..." (если строка была усечена). Если последний символ усеченной строки является пробелом – удалить его и добавить заполнитель.

Пример:

truncate("39 новых фич, которые будут доступны в Java 12") //39 новых фич, ко...

truncate("39 новых фич, которые будут доступны в Java 12", 9) //39 новых…

truncate("A ", 3) //A

Обратите внимание, если строка заканчивается пробелами, часть из которых нужно усекать, то все пробелы в конце строки удаляются.

1. (на оценку хорошо) Пользователь вводит некоторый текст и выбирает один из способов изменения регистра введенного текста:
   1. Как в предложениях
   2. все строчные
   3. ВСЕ ПРОПИСНЫЕ
   4. Начинать С Прописных
   5. иЗМЕНИТЬ РЕГИСТР

Название методов должно отражать их действия.

1. (на оценку отлично) Для реализации задачи на изменение регистра использовать шаблон проектирования Стратегия.