Установка пакета

1. В папку с проектом перемещаем файлы CSV.java и Tuple.java (в моем случае основной файл называется csv\_test.java)
2. Компилируем файлы библиотеки с помощью

  


1. В проекте подключаем с помощью



1. Компилируем основной файл и запускаем его с помощью



Пример использования 1 (запись словаря):

import csv.CSV;

import csv.Tuple;

import java.util.\*;

public class csv\_test

{

    public static void main(String[] args)

    {

        CSV test = new CSV(); // создание объекта для записи

        Map<String, List<Double>> data = new HashMap<String, List<Double>>(); // создание словаря с данными, тип --> Map<key, List<value>> <--

        List<Double> time = new ArrayList<Double>(); // массив с замерами времени выполнения

        time.add(2.3);

        time.add(3.2);

        List<Double> count = new ArrayList<Double>(); // массив с количеством данных

        count.add( 3.0);

        count.add( 2.0);

        data.put("Count", count); // добавляем по ключу "Count" количество данных

        data.put("Time(ms)", time); // добавляем по ключу "Time(ms)" замеры времени

        test.writeDict("dict.csv", data); // передаем в функцию название файла и словарь с данными

    }

}

Пример использования 2 (запись списка строк):

import csv.CSV;

import csv.Tuple;

import java.util.\*;

public class csv\_test

{

    public static void main(String[] args)

    {

    // класс безопасен, есть различные проверки, при использовании, как на примере, не сломается

        List<Tuple<Integer, Double>> data = new ArrayList<Tuple<Integer, Double>>(); // <- данные в таком виде --> List<Tuple<T1, T2>> <--

        data.add(new Tuple<Integer, Double>(10, 2.3));

        data.add(new Tuple<Integer, Double>(20, 5.6));

        data.add(new Tuple<Integer, Double>(30, 10.1));

        CSV test = new CSV(); // <- создали объект

        test.fillFile("new\_file.csv", "Count;Time", data);

        //ИЛИ

        test.open("new\_file.csv"); // <- открыли нужный файл (безопасно, проверка на уже открытый файл или на ошибку создания)

        test.writeHeader("Count;Time"); // <- записали нужный заголовок (безопасно, не дублирует при многократном вызове)

        test.write(data); // <- записали данные

        test.close(); // <- закрыли (безопасно, не даст закрыть файл, если еще не открыли)

    }

}