|  |  |
| --- | --- |
| 01. | Copy project folder **"HW\_29"**to **"HW\_30"**and import it to Eclipse. |
| 02. | Modify artifactId & name: |
| 03. | Change default JRE to JDK 1.8 162 |
| 04. | Create the following text file: **/Users/jsmith/input.txt                              C:\>input.txt** |
|  | |  |  | | --- | --- | | 01 | **John Smith** | | 02 | **Mike Moore** | | 03 | **Alex Loomis** |   . |
| 05. | Create the following properties file: **/Users/jsmith/input.properties          C:\>input.properties** |
|  | |  |  | | --- | --- | | 01 | **f\_name = John** | | 03 | **l\_name = Smith** |   . |
| 06. | Create the following csv file: **/Users/jsmith/input.csv                               C:\>input.csv** |
|  | |  |  | | --- | --- | | 01 | **John,Smith** | | 02 | **Mike,Moore** | | 03 | **Marco,Polo** |   . |
| 07. | Create the following Java classes (4): |
|  | **Input\_Text\_FR.java**  **Input\_Text\_Scanner.java**  **Input\_PF.java**  **Input\_CSV.java**  ***Если после запуска jar-with-dependencies выдает ошибку что не может найти файл, с которого парсается дата, значит в коде не указан динамический запуск, а если запускать хардкоднуто тогда не нужно в командной строке указывать путь к файлу, так как он указан в коде.***  **.** |
|  | **Class: Input\_Text\_FR.java**  **Ищет значения построчно, каждое значение должно быть на каждой строке. Каждая переменная выводится на отдельной строке. Переменная String full name = “Vasya Ryjov” будет на одной строке.** |
|  | **java -cp HW\_30-1.0-jar-with-dependencies.jar core.Input\_Text\_FR ~/input.txt**    **Class: Input\_Text\_Scanner.java**  **Works with no problem if a path is hardcoded. File should be in the same folder as “target”, “src” and “bin” folders are. Чтобы сканнер нашел файл нужно чтобы этот файл был в той же папке откуда мы его и запускаем и путь к этому файлу должен быть захардкоден. Сканнер ищет по пробелу: s.next() – ищет по пробелу. Например если хочешь вывести 3 слова на одной строке, надо заассайнать метод s.next() к каждой переменной. Сколько методов s.next() столько будет значений на одной строке. Вот к примеру добавили переменную String middle\_name = s.next(); и теперь на одной строке появилось отчество.** |
|  | **$ java -cp HW\_30-1.0-jar-with-dependencies.jar core.Input\_Text\_Scanner ~/input.txt** |
|  | **Class: Input\_PF.java**  **В file.properties файле все переменные должны быть уникальны. Расширения файла не имеет значения, но принято называть file.properties. Существует 2 вида проперти файла External and Internal. Если нужно что-то скрыть, тогда лучше пользовать External. Преимущество Internal что он ищет по всем папкам этот файл, но минус что если найдет а этих файлов будет множество, то он возьмет самый первый и будет работать с ним** |
|  | **$ java -cp ./target/HW\_29-1.0.jar core.Input\_PF ~/input.properties**    **Class: Input\_CSV.java**  **Разделяет значения запятой или любым другим знаком, которым нужно, только в коде нужно прописать какой именно знак при помощи RegEx.** |

**$ java -cp ./target/HW\_29-1.0-jar-with-dependencies.jar core.Input\_Text\_CSV ~/input.csv**

