Упрощенный парсер CSV

## Subtask #1 - print tuple

Используя рекурсивные шаблоны реализовать оператор для печати std:tuple:

***…..  
auto operator****<<(std::basic\_ostream<Ch, Tr>& os, std::tuple<Args...>* ***const****& t)****…..***

## Subtask #2 - simple CSV parser

### CSV format

Табличные данные могут быть представлены как текстовый файл с разделителем ‘\n’ между строками и символом ‘,’ для разделения ячеек внутри строки. Считаем что данные символы не встречаются внутри данных.

### CSVParser

Написать класс делающий возможным следующую потоковую работу с CSV:

***int*** *main()*

*{*

*ifstream file(****"test.csv"****);*

*CSVParser<****int****, string> parser(file, 0 /\*skip first lines count\*/);*

***for*** *(tuple<****int****, string> rs : parser) {*

*cout<<rs<<endl;*

*}*

*}*

Потоковая обработка подразумевает lazy (ленивое) чтение строк. Таким образом необходимо реализовать <http://en.cppreference.com/w/cpp/concept/InputIterator> для чтения данных в CSV файле.

## Subtask #2 - improved CSV parser

Добавить следующие возможности:

1. Поддержка экранирования данных
2. Конфигурация парсера: что считать разделителем между строками и колонками и что за символ использовать для экранирования (по умолчанию двойные кавычки)
3. Обработка ошибок: выкидывать исключение с информаций о месте в файле (строка, колонка) где произошла ошибка разбора данных