



CSV

DeepDive Week 0 Day 2



Материалы и Ресурсы

Сегодня день посвящен:

- CSV - <https://ru.wikipedia.org/wiki/CSV>
- Сериализации и десериализации данных - <https://ru.wikipedia.org/wiki/Сериализация>
- Записям ([records](#))
- Паттерну [Builder](#)
- [Data Classes and Sealed Types](#) for Java
- [Data Classes](#) for Java

Важно! Для данного задания необходимо использовать Java 15!

Для начала мы реализуем простой record-класс, который будет хранить структуру CSV, а также *Builder* для создания экземпляров этого класса.

CSV

```
package academy.kovalevskyi.javadeepdive.week0.day2;

public record CSV(String[] header, String[][] values) {

    public static class Builder {

        public Builder header(String[] header) {
            // TODO
        }

        public Builder values(String[][] values) {
            // TODO
        }

        public CSV build() {
            // TODO
        }
    }

    public boolean withHeader() {
        // TODO
    }

    @Override
    public boolean equals(Object o) {
        // TODO
    }

    @Override
    public int hashCode() {
        // TODO
    }
}
```

После необходимо реализовать класс, который помогает сериализации и десериализации CSV на/из файловой системе.

CSVHelper

```
package academy.kovalevskyi.javadeepdive.week0.day2;
import academy.kovalevskyi.javadeepdive.week0.day0.StdBufferedReader;
import java.io.*;
import java.util.ArrayList;

public class CSVHelper {

    public static CSV parseFile(String fileName) throws FileNotFoundException {
        // TODO
    }

    public static CSV parseFile(String fileName, boolean withHeader, char
delimiter) throws FileNotFoundException {
        // TODO
    }

    public static void writeCsv(String fileName, CSV csv, char delimiter) throws
IOException {
        // TODO
    }
}
```

Обратите внимание! Для чтения файл должен использоваться ранее созданный *StdBufferedReader*, что может привести к тому, что будут найдены новые баги буфера.