

Королев Даниил Алексеевич

+7 (981) 722-11-96

daniilkorolev00@gmail.com

github.com/Danielto1404

Технические навыки

Языки программирования: **Java, Swift, C++**

также знаком с **Python, Haskell**

Frameworks: **Junit,**

ios: Lottie, Firebase, MapBox, GoogleMaps, YouTube API

Образование

Университет ИТМО, Санкт-Петербург

Факультет информационных технологий и программирования

01.03.02 Прикладная математика и информатика

2018—2022

Президентский физико-математический лицей № 239, Санкт-Петербург

Проекты

Jar-Implementor [Java] [ссылка на код \(https://github.com/Danielto1404/University/tree/master/java-advanced/04-06-JarImplementor\)](https://github.com/Danielto1404/University/tree/master/java-advanced/04-06-JarImplementor)

1. Реализован класс `Implementor`, который генерирует реализации классов и интерфейсов.
 - Аргумент командной строки: полное имя класса/интерфейса, для которого требуется сгенерировать реализацию.
 - В результате работы генерируется java-код класса с суффиксом `Impl`, расширяющий (реализующий) указанный класс (интерфейс).
 - Сгенерированный класс не должен быть абстрактным.
 - Методы сгенерированного класса должны игнорировать свои аргументы и возвращают значения по умолчанию.

HelloNonBlockingUDP [Java] [ссылка на код \(https://github.com/Danielto1404/University/tree/master/java-advanced/12-HelloNonblockingUDP\)](https://github.com/Danielto1404/University/tree/master/java-advanced/12-HelloNonblockingUDP)

1. Разработаны два класса: клиент и сервер, взаимодействующие по UDP.
2. Класс `HelloUDPClient` должен отправлять запросы на сервер, принимать результаты и выводить их на консоль.
 - Аргументы командной строки:
 1. имя или ip-адрес компьютера, на котором запущен сервер;
 2. номер порта, на который отсылать запросы;
 3. префикс запросов (строка);
 4. число параллельных потоков запросов;
 5. число запросов в каждом потоке.
 - Запросы должны одновременно отсылаться в указанном числе потоков. Каждый поток должен ожидать обработки своего запроса и выводить сам запрос и результат его обработки на консоль. Если запрос не был обработан, требуется послать его заново.
 - Запросы должны формироваться по схеме `<префикс запросов><номер потока>_<номер запроса в потоке>`.
3. Класс `HelloUDPNonBlockingServer` должен принимать задания, отсылаемые классом `HelloUDPNonBlockingClient` и отвечать на них.
 - Аргументы командной строки:

1. номер порта, по которому будут приниматься запросы;
 2. число рабочих потоков, которые будут обрабатывать запросы.
- Ответом на запрос должно быть Hello, <текст запроса>.
 - Если сервер не успевает обрабатывать запросы, прием запросов может быть временно приостановлен

IRLKiller [Swift] [ссылка на код \(https://github.com/B1boid/IRLKiller\)](https://github.com/B1boid/IRLKiller)

Подробное описание проекта и видео к проекту можно посмотреть перейдя по ссылке указанной выше.

Больше проектов можно посмотреть на моем гитхабе: <https://github.com/Danielto1404/University>