Обзор подходящих ЯП.

Для реализации программы мониторинга сети нужно было выбрать подходящий язык программирования. Выбирал из нескольких языков:

1. Golang
2. Java
3. Python
4. Node.js
5. Clojure

**Golang** - компилируемый язык программирования от компании Google с хорошей поддержкой многопоточности встроенной в сам язык. Также в него встроен сборщик мусора, что сильно облегчает написание программ.

Плюсы:

* Компилируемый - все зависимости проекта компилируется в один исполняемый файл.
* Встроенный сборщик мусора - не нужно заботиться об освобождении памяти, сборщик мусора все делает за вас.
* Большое количество библиотек с открытым исходным кодом.
* Довольно большое сообщество - есть кому задать вопросы.
* Высокая скорость разработки

Минусы:

* При переносе проекта из одной операционной системы в другую необходимо перекомпилировать проект.
* Из-за санкций нет доступа к огромному количеству библиотек(golang.org)
* Нет встроенной системы зависимостей.

**Python** - динамический, интерпретируемый, высокоуровневый язык программирования, поддерживающий множество парадигм.

Плюсы:

* Динамический - подходит для быстрого прототипирования.
* Огромное количество качественных библиотек на все нужды
* Интерпретируемый - легко переносить проекты на другие операционные системы без перекомпиляции.
* Большое и дружелюбное сообщество, готовое прийти на помощь в любой момент.
* Возможность писать в функциональном стиле.

Минусы:

* Низкое быстродействие.
* Невозможно модифицировать встроенные классы
* Глобальная блокировка интерпретатора (GIL) - в одно время в одном процессе python может выполняться только один поток.

**Java** - объектно ориентированный язык программирования с сильной типизацией от Sun Microsystems.

Плюсы:

* Компиляция программ в байт-код - повышение производительности, в сравнении с интерпретируемыми языками.
* Переносимость - виртуальная машина java реализована под большинство операционных систем.
* Большое количество и хорошее качество библиотек.
* Сборка мусора.

Минусы:

* Повышенное потребление оперативной памяти.
* Низкая выразительность - раздутие кода.
* Необходимость установки виртуальной машины java для запуска программы.

**Node.js** - платформа, построенная поверх javascript движка V8, транслирующего javascript в машинный код.

Плюсы:

* Клиентский и серверный код можно писать на одном и том же языке.
* Большое количество библиотек.
* Платформа построена на асинхронной событийной модели - удобно при работе с сетью.

Минусы:

* Многие библиотеки сомнительного качества.
* Сложность обработки исключений.

**Clojure** - динамический язык программирования общего назначения, сочетающий в себе все преимущества интерактивной разработки с эффективной и надежной инфраструктурой для многопоточного программирования. Clojure - это диалект лиспа, разделяющий с ним философию кода как данных и имеющий мощную макросистему.

Плюсы:

* Компилируется в байт-код java - все преимущества java, а также возможность напрямую использовать java библиотеки.
* Функциональный стиль
* Интерактивная разработка в REPL-цикле.
* Неизменяемые структуры данных.
* В связке с Clojurescript, который компилируется в javascript получаем один язык программирования на клиенте и сервере.
* Великолепная поддержка многопоточности.

Минусы:

* Небольшое комьюнити.

Написав прототипы на каждом языке я пришел к выводу использовать в своем проекте Clojure.