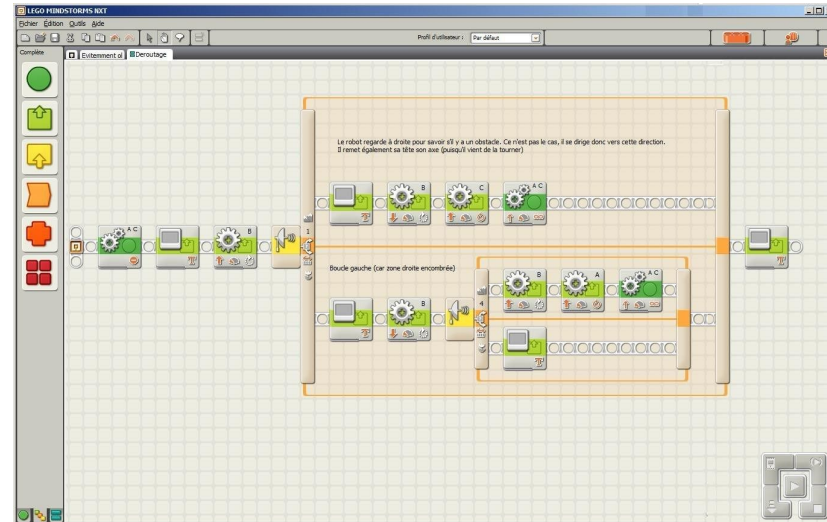
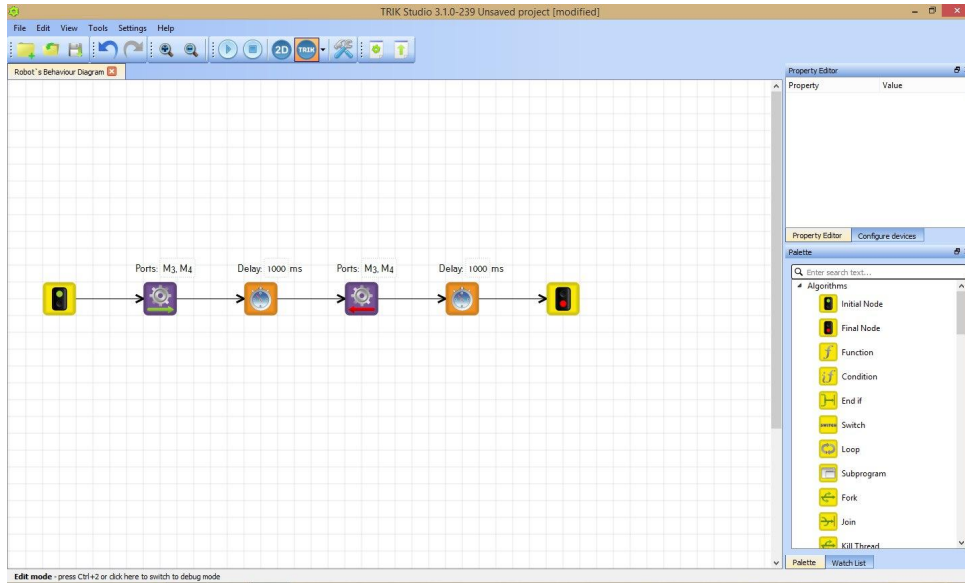


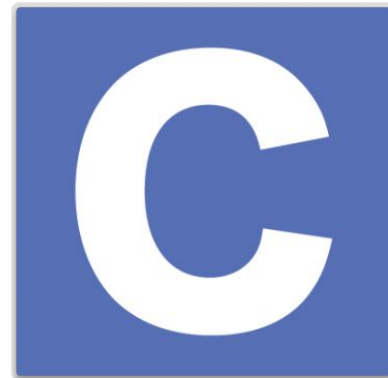
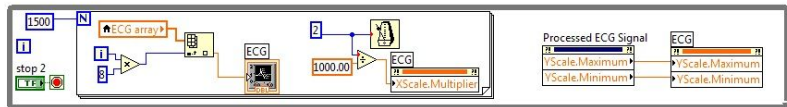
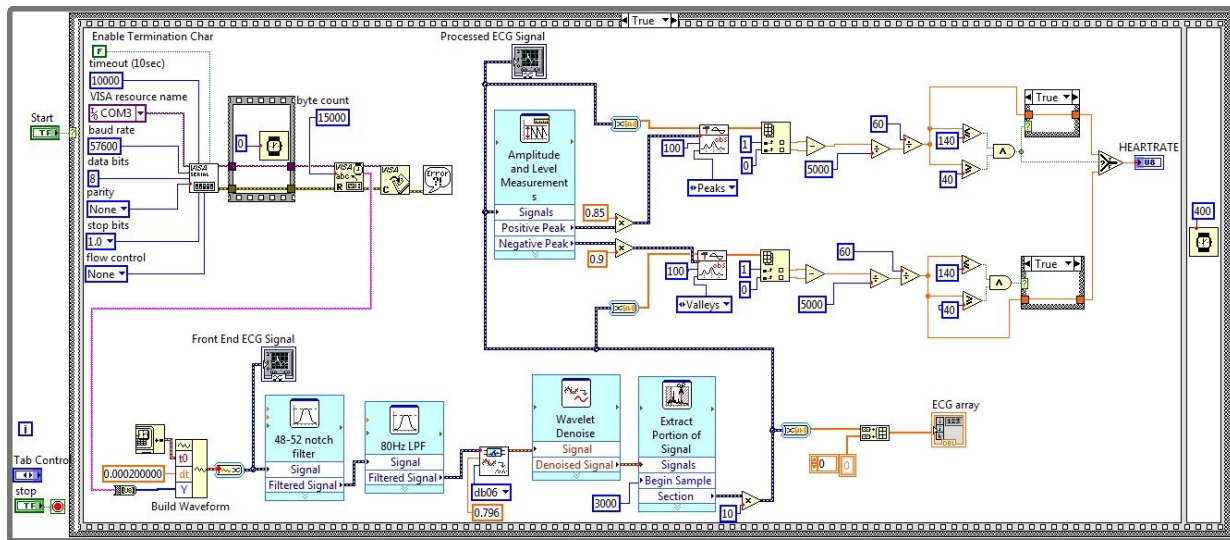
# Программирование роботов ТРИК с помощью JVM

Белков Роман Владимирович, 371  
Научный руководитель: ст. преп. Я. А.  
Кириленко

# Введение



# Введение



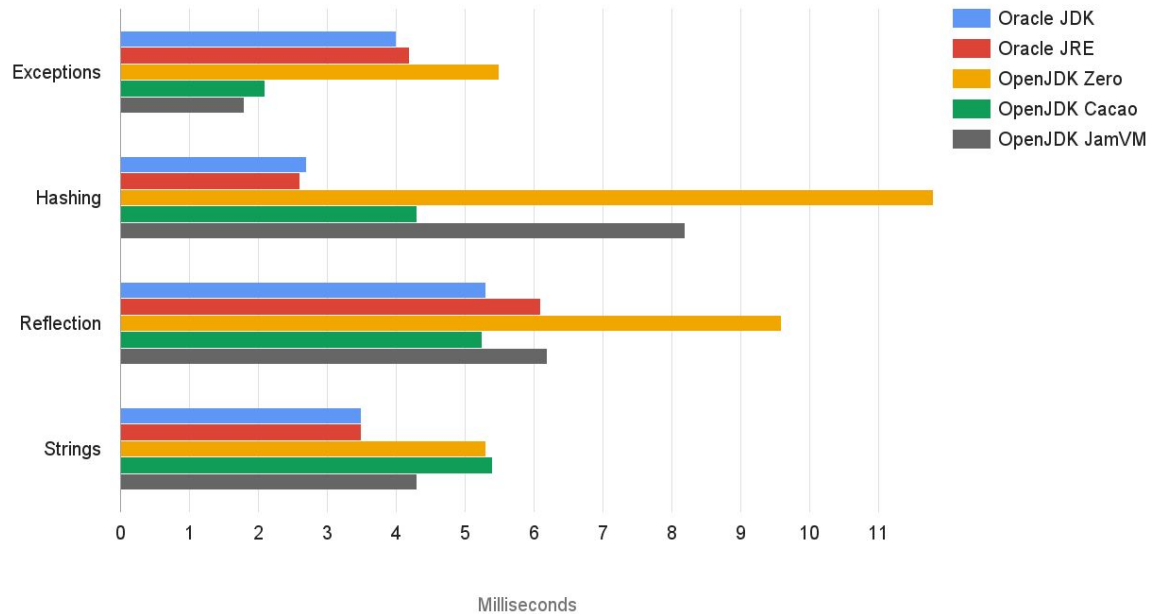
# Цель

Реализовать возможность программирования  
роботов на языке Java

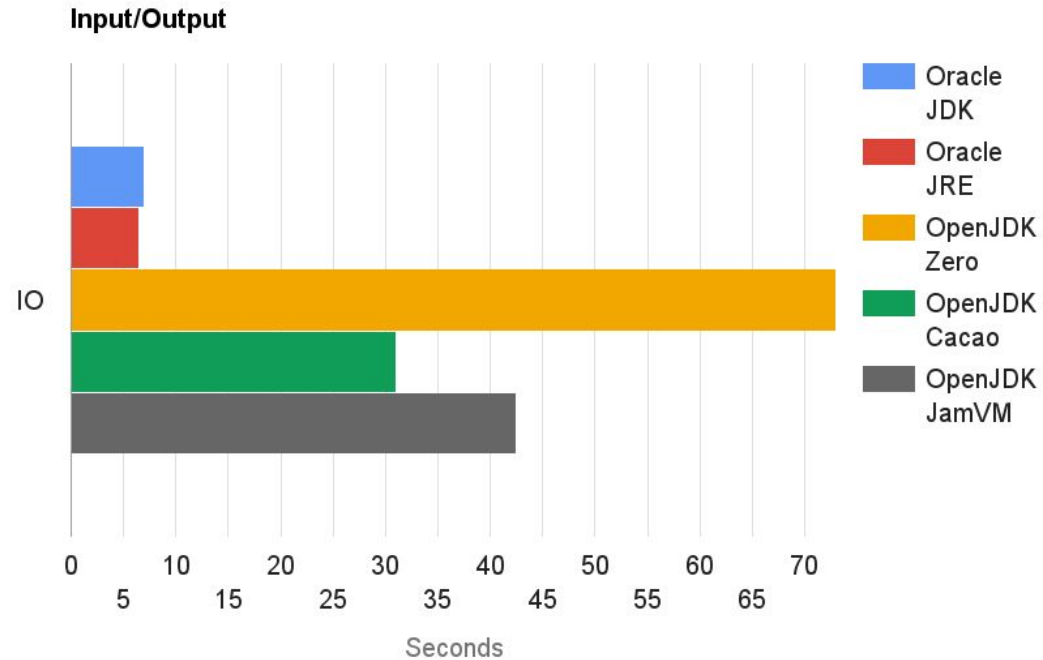
# Задачи

- Развернуть JDK на ТРИК
- Создать библиотеку времени исполнения
  - Поддержка основных сенсоров ТРИК
  - Событийно-ориентированное программирование
  - Высокий уровень абстракции

# Обзор технологий. Выбор JDK

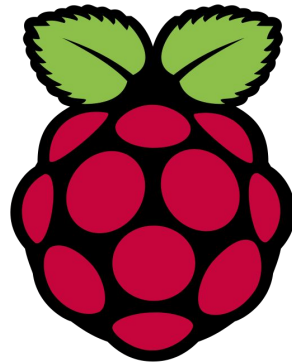


# Обзор технологий. Выбор JDK



# Обзор технологий. Выбор JDK

Алгоритм	Решето Аткина	Операции с плавающей запятой
Отношение времени работы OpenJDK к Oracle JDK	14	13,6





# Выбор языка для библиотеки

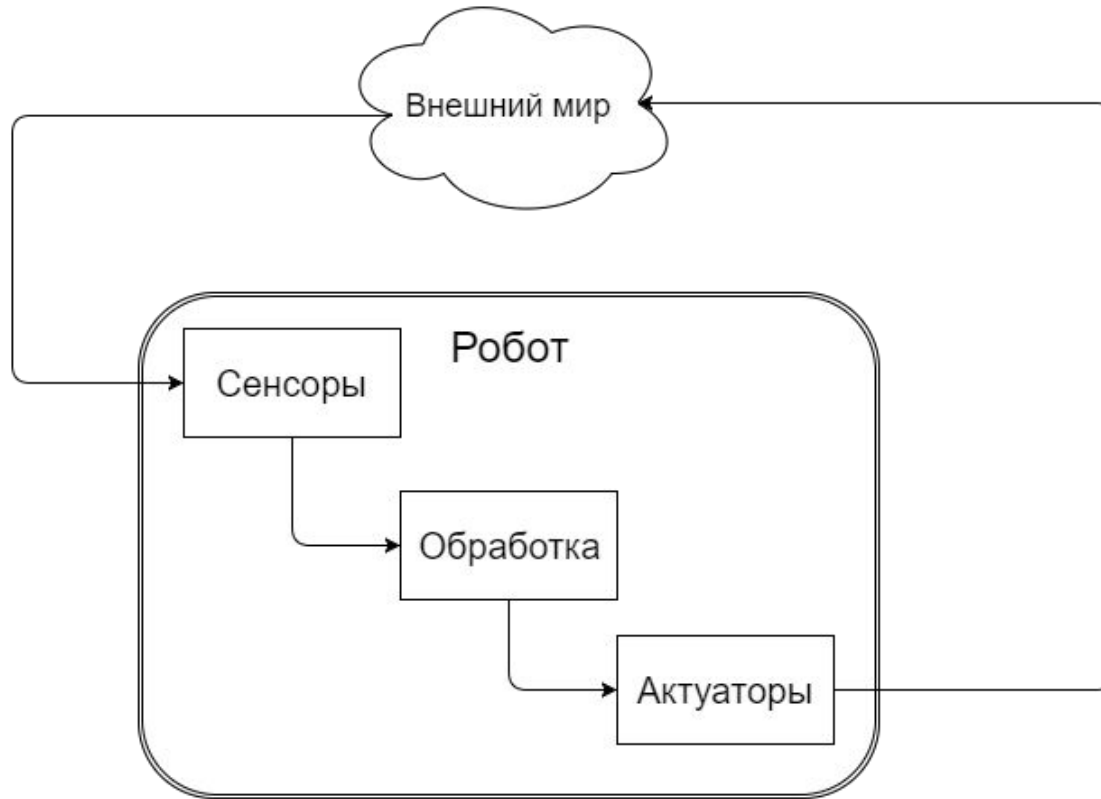
Язык	Java	Scala	Groovy	Clojure	Kotlin
Объектно-ориентированный	+	+	+	-	+
Функциональный	-	+	-	+	+
Статическая типизация	+	+	-*	-	+
Хорошее взаимодействие с Java		-	+	-	+
Простой синтаксис	+	-	+	-	+

# Опыт работы с Kotlin

- Безопасные типы
- Методы расширения
- Конструкция when



# Событийно-ориентированное программирование



# Событийно-ориентированное программирование

```
fun limitServo(x: Int) = limitAbs(MAX_SERVO_ANGLE, x)  
  
val aimX =  
    locationStream  
        .scan(0, { acc, loc -> limitServo(acc - updatePositionX(loc.x)) })  
        .subscribe(servoX)  
  
val aimY =  
    locationStream  
        .scan(0, { acc, loc -> limitServo(acc - updatePositionY(loc.y)) })  
        .subscribe(servoY)
```



# Поддержанные устройства

- Силовой мотор с энкодером
- Угловой сервопривод
- Аналоговые датчики
- Цифровые датчики
- Видеосенсоры
- Гироскоп
- Акселерометр
- Кнопки действий
- Управление с пульта

# Документация и примеры

- Описание методов
- Примеры:
  - сегвей
  - “охотник”
  - сетевое взаимодействие по TCP

# Результаты

- Развёрнута Java SE на ТРИК
- Готова библиотека времени исполнения
  - Поддержка событийно-ориентированного программирования
- Демо-программы и документация