

Санкт-Петербургский государственный университет

Математическое обеспечение и
администрирование информационных систем

Кафедра системного программирования

Коекин Ярослав Алексеевич

Взаимодействие с TRIK роботом с
помощью сред разработки Visual Studio
Code и Pycharm

Курсовая работа

Научный руководитель:
ст. преп. Я. А. Кириленко

Санкт-Петербург

2019

Содержание

Введение	3
1 Обзор	5
1.1 Существующие решения	5
1.2 Используемые технологии	6
2 Описание решения	7
2.1 Реализация расширения для PyCharm	7
2.2 Реализация извлечения информации о переменных .	9
3 Заключение	11
Список используемой литературы	12

Введение

Команда преподавателей и студентов СПбГУ TRIK создала контроллер, который позволяет конструировать различные робототехнические модели. Возможности по управлению такими моделями предоставляет среда программирования TRIK Studio. Она позиционируется как среда для обучения основам программирования и кибернетики. TRIK Studio предлагает учиться управлять роботами посредством диаграмм, задавая поведение робота с помощью блоков.

Способ задания поведения с помощью диаграмм позволяет абстрагироваться от непосредственного написания кода на сложных текстовых языках, что облегчает процесс программирования. Таким образом, пользователями TRIK Studio могут стать даже дети младшего школьного возраста.

Однако, такой подход не сможет обучить пользователя взаимодействовать с реальными языками программирования. Помимо этого, TRIK Studio не является популярной средой разработки, что исключает процесс накопления опыта работы с такими известными и широкопрофильными средами разработки как, например, с PyCharm и Microsoft Visual Studio Code при обучении программированию на TRIK Studio.

Возникает задача – реализовать возможность задавать поведение TRIK робота (робототехническая модель, основанная на TRIK-контроллере) средствами популярных сред разработки на реальных текстовых языках.

На данный момент, TRIK-контроллер [1] может воспринимать скрипты, написанные на таких языках программирования как JavaScript и Python.

Таким образом, цель настоящей работы – разработка и создание расширений для Visual Studio Code и PyCharm, которые бы

позволили взаимодействовать с TRIK роботом напрямую из данных сред.

Для достижения данной цели были поставлены следующие задачи:

Реализовать возможность средствами Pycharm:

- запускать программу, написанную на языке Python на TRIK роботе;
- останавливать выполнение программы на роботе;
- отправлять файл на робота;
- получать информацию о соединении.

Помимо этого, реализовать возможность средствами Microsoft Visual Code:

- получать информацию о состоянии переменных в момент выполнения программы, написанной на языке JavaScript.

1 Обзор

1.1 Существующие решения

TRIK Studio

TRIK Studio – интуитивно понятная среда программирования, которая позволяет взаимодействовать с роботами с помощью последовательности блоков. В среде реализовано программирование роботов TRIK, а также Lego Mindsorms NXT 2.0 и EV3, но возможности таких роботов сильно ограничены в сравнении с TRIK. TRIK Studio базируется на QReal [2] – это DSM-платформа¹, которая позволяет создавать визуальные языки программирования.

Отличительная особенность данной среды заключается в возможности интерактивно имитировать процесс моделирования управления роботом.

Создание программ в данной среде основано на диаграммах, что затрудняет процесс использования текстовых языков программирования.

EV3 Python Extension

EV3 Python – библиотека, позволяющая управлять Lego EV3 роботом с помощью языка программирования Python. Этот инструмент имеет поддержку в виде расширения для Microsoft Visual Studio Code [3]. Таким образом, существует возможность управлять робототехническим комплексом Lego EV3 из среды разработки Visual Studio Code, задавая поведение с помощью Python.

Данный инструмент не подходит, т. к. он не позволяет взаимодействовать с TRIK роботом. Более того, процесс установки взаимодействия между Visual Studio Code и Lego EV3 является нетривиальным и состоит из множества этапов.

¹Domain-Specific Modeling

1.2 Используемые технологии

Java

Java – объектно-ориентированный язык программирования. Используется для создания расширений среды программирования PyCharm наряду с другим языком программирования – Kotlin. При создании расширения был выбран именно язык Java, т. к. он является одним из самых популярных языков программирования [4]; помимо этого, большинство расширений для PyCharm создано именно на Java.

Для сборки Java-проекта в официальной документации JetBrains по созданию плагинов советуют использовать систему автоматической сборки Gradle. А для конфигурации проекта используется диалект Java Groovy.

TypeScript

TypeScript – язык программирования, позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript. В официальной документации Visual Studio Code авторы утверждают, что именно TypeScript является наиболее удобным языком для разработки расширений.

2 Описание решения

2.1 Реализация расширения для PyCharm

При реализации задачи была разработана архитектура, использующая паттерн Model-View-Presenter, который позволяет отделить данные, логику взаимодействия и визуализацию друг от друга. Получившаяся архитектура в виде диаграммы классов представлена на рисунке 1.

Класс View имеет инструменты для взаимодействия с пользователем. Здесь обрабатываются действия пользователя, а также запускаются основные механизмы класса Presenter, активируя основные возможности расширения: послать файл роботу, запустить программу, остановить выполнение программы на работе и др. Класс Model хранит информацию об адресе текущего соединения и о данных текущего открытого в среде программирования PyCharm файла.

Класс Connection отвечает за связь между устройством и расширением. От этого класса наследуется класс RobotConnection, который хранит методы, специфичные для взаимодействия с TRIK роботом.

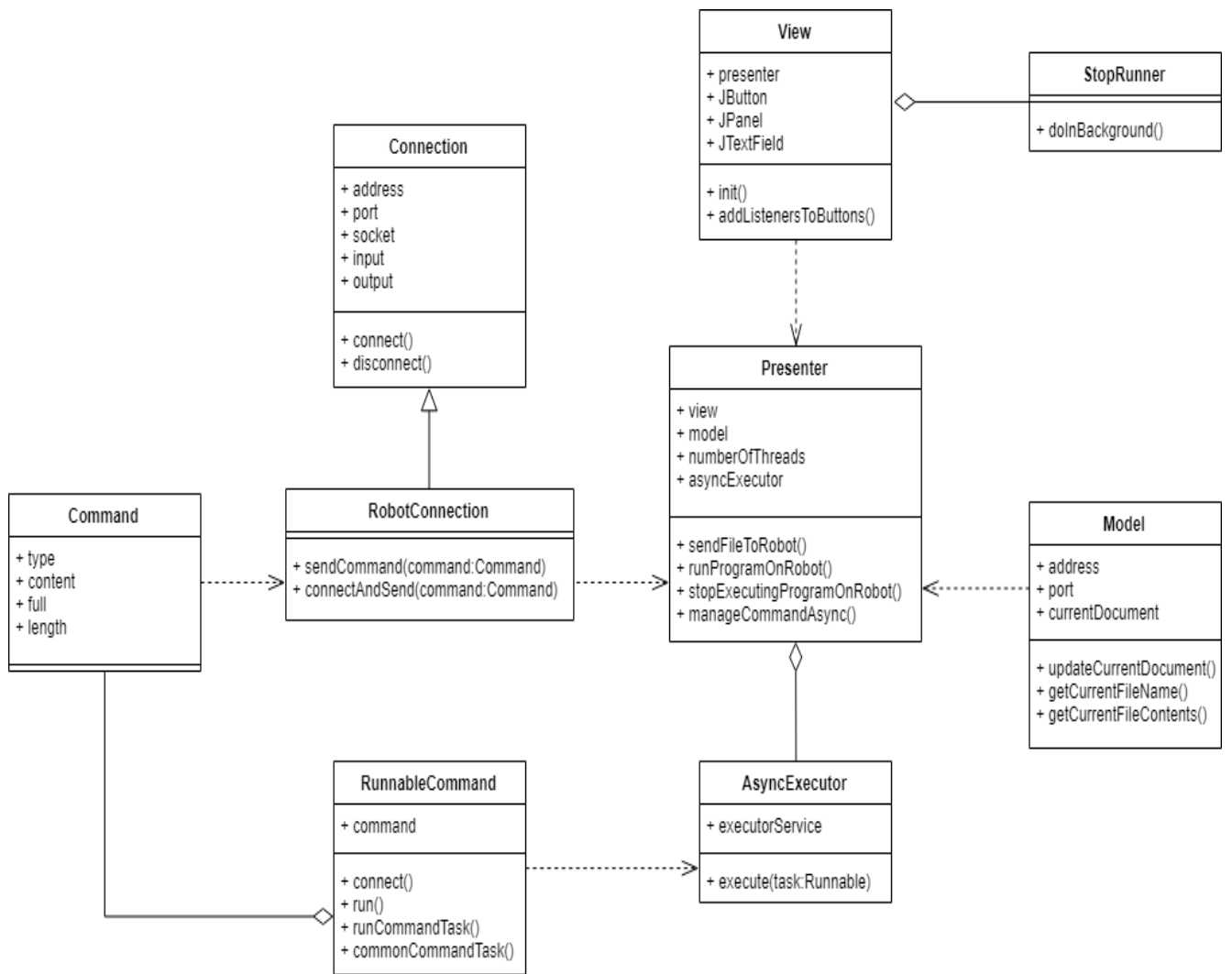


Рис. 1: Диаграмма классов расширения PyCharm.

2.2 Реализация извлечения информации о переменных

В ходе работы были реализованы классы, выделенные на рисунке 2. Эти классы – лишь часть общей структуры расширения для Microsoft Visual Studio Code.

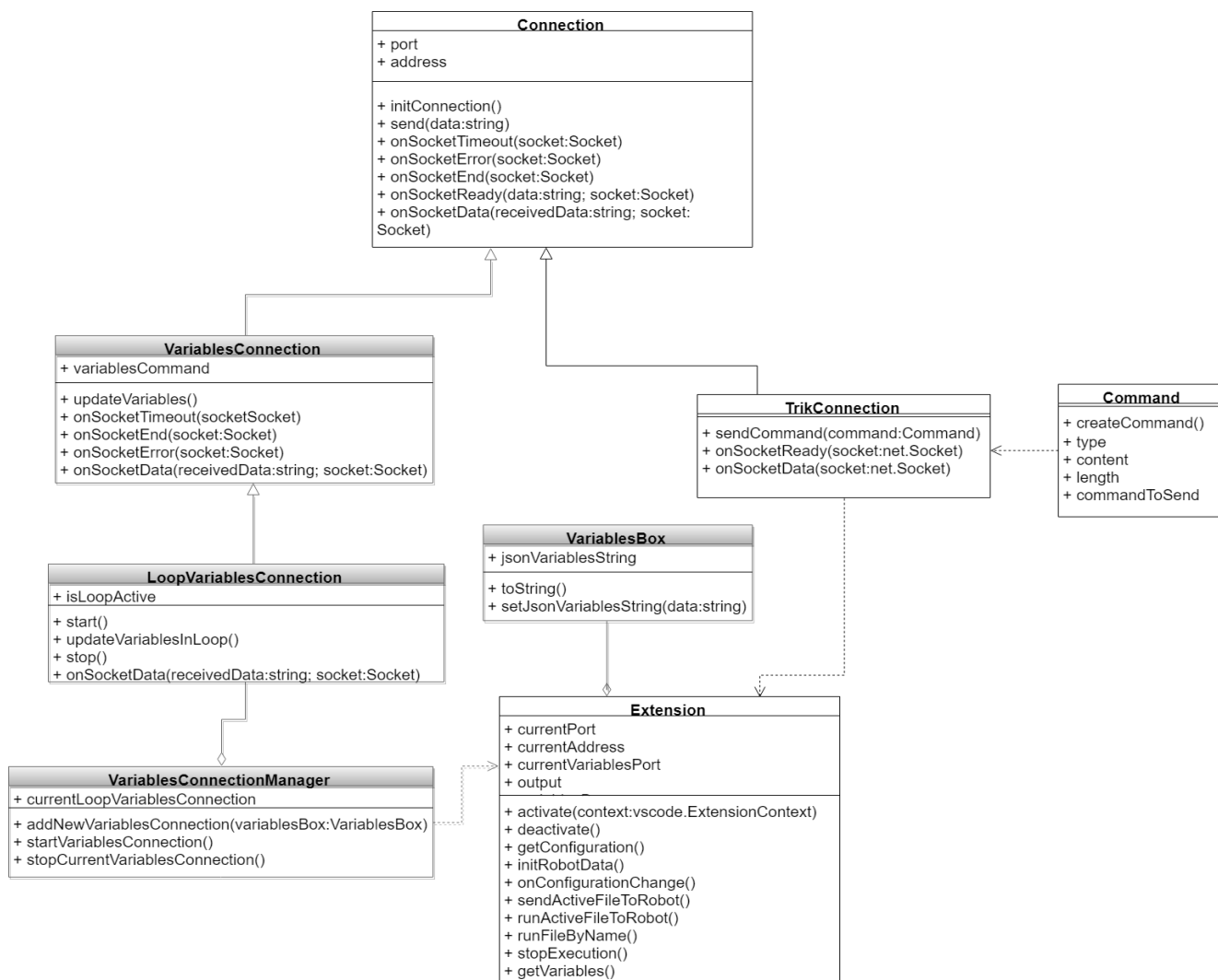


Рис. 2: Диаграмма классов расширения VsCode.

VariablesConnection – класс, наследуемый от класса Connection. Реализует специфичные методы для связи с сервером переменных

TRIK работа. Сервер переменных предоставляет информацию о значениях переменных во время выполнения очередной программы, написанной на JavaScript на TRIK работе. Средствами класса VariablesConnectionManager обеспечивается корректное извлечение значений переменных.

3 Заключение

В рамках данной курсовой работы сделано:

Для расширения PyCharm реализованы возможности:

- запускать программу, написанную на языке Python на TRIK Роботе;
- останавливать выполнение программы на работе;
- отправлять файл на работа;
- получать информацию о соединении.

Для расширения Microsoft Visual Studio Code реализована:

- возможность получать информацию о состоянии переменных в момент выполнения программы, написанной на языке JavaScript.

Список используемой литературы

[1] Wiki страница trikRuntime.

<https://github.com/trikset/trikRuntime/wiki>

(дата обращения: 24.05.2019)

[2] Wiki страница QReal.

<https://github.com/qreal/qreal/wiki>

(дата обращения: 26.05.2019)

[3] EV3 extension for Microsoft Visual Studio Code.

[https://sites.google.com/site/ev3python/
setting-up-vs-code](https://sites.google.com/site/ev3python/setting-up-vs-code)

(дата обращения: 01.06.2019)

[4] TIOBE Index for May 2019.

<https://www.tiobe.com/tiobe-index/>

(дата обращения: 01.06.2019)