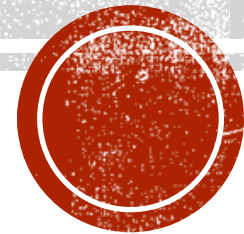


# ЯЗЫК ПРОГРАММИРОВАНИЯ ДЛЯ ПРАКТИКУМОВ ПО РАСПРЕДЕЛЁННЫМ СИСТЕМАМ МИКРОКОНТРОЛЛЕРОВ

Богачёв Алексей Иу-5



# ОСНОВНАЯ ИДЕЯ

- Создание простого в освоении императивного языка программирования
- Возможность упростить для понимания идеи распределённых систем и многопоточных вычислений за счёт использования модели акторов
- Наглядная демонстрация возможностей языка и модели акторов на примере распределённых систем микроконтроллеров
- Упрощение освоения других языков программирования использующих обмен данными через сообщения, например **Scala**, **Erlang**, **Objective C**, **Smalltalk**.
- Возможность наглядного изучения работы виртуальной машины



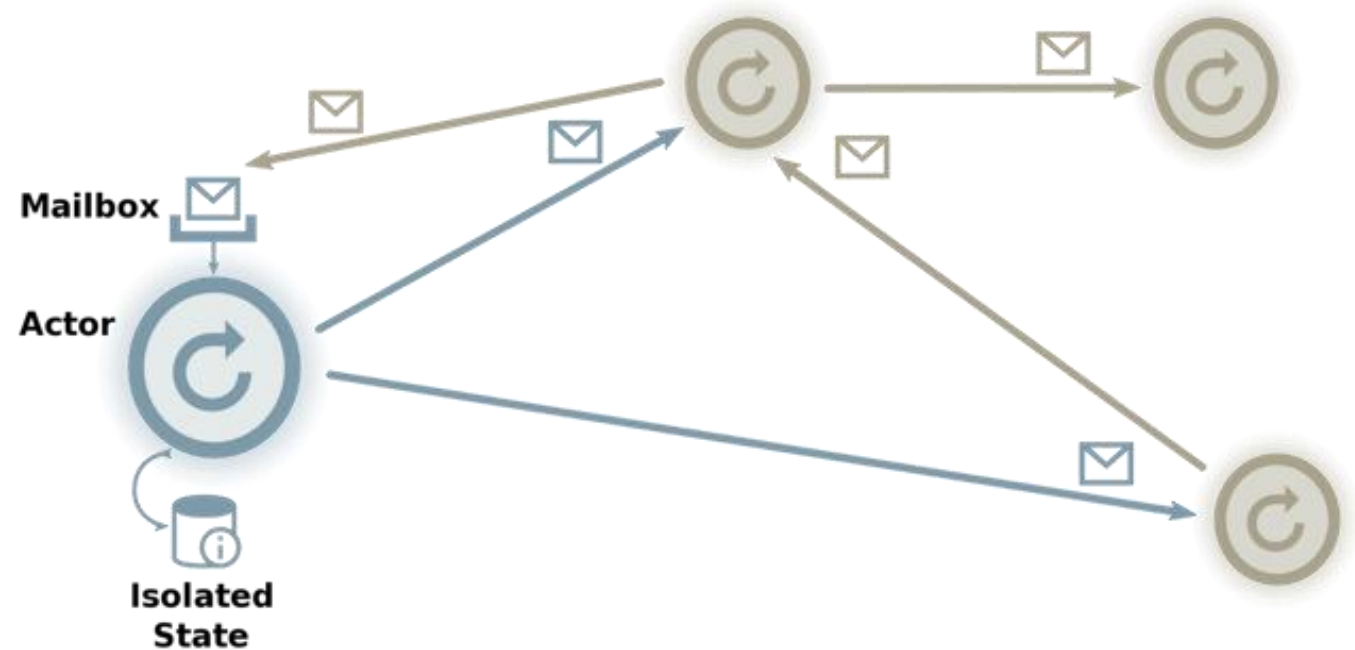
# ОСНОВНЫЕ КОНЦЕПЦИИ ЯЗЫКА

- Использование модели акторов
- Использование специальной виртуальной машины
- Работа с функциями как с объектами первого класса
- Небольшая но универсальная стандартная библиотека
- Простота для освоения
- Однозначность и предсказуемость поведения
- Однозначная транслируемость программ в байт-код и обратно



# МОДЕЛЬ АКТОРОВ

- Актор – сущность которая может отправить конечное число сообщений другим акторам
- Создать конечное число новых акторов
- Выбрать тип поведения для следующего сообщения
- В данном случае – акторы являются процессами занимающимися обработкой сообщений



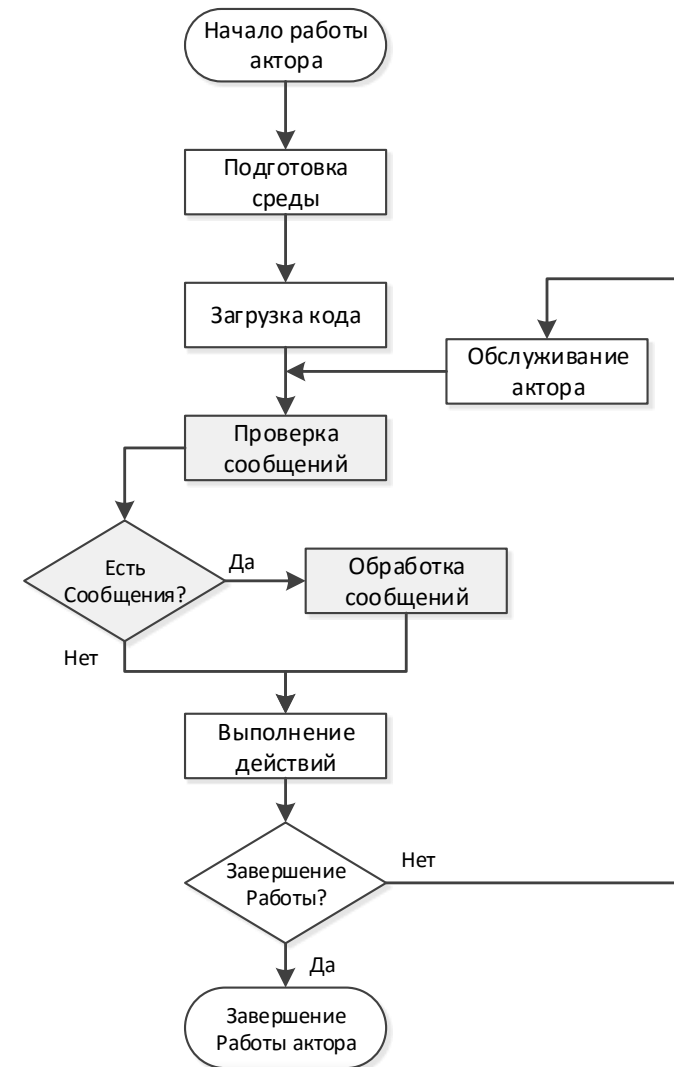
# БАЙТ-КОД И ДАННЫЕ

- Язык использует статическую типизацию данных
- Данные хранятся вместе с сигнатурами, позволяющими определить их тип
- Сигнатуры позволяют реализовать полиморфизм на основе мультиметодов
- Сигнатуры упрощают сериализацию и передачу данных по сети, а также дальнейшую обработку этих данных
- Байт-код имеет структурированный формат, позволяющий работать с функциями как с объектами первого класса, переопределять и передавать их по сети



# МЕХАНИЗМ ОБМЕНА СООБЩЕНИЯМИ

- Взаимодействие акторов осуществляется с помощью механизма сообщений
- Каждый актор в той или иной форме является обработчиком сообщений
- Сообщения позволяют применять методологию ООП при работе с акторами



# МЕХАНИЗМ ЗАМЕНЫ ПРОГРАММНОГО КОДА

- Позволяет обновлять программный код без остановки системы, переопределять отдельно взятые функции
- Позволяет экономить используемую оперативную память подгружая только необходимые функции из разделяемых библиотек
- Позволяет динамически подгружать в память необходимые в данный момент функции, как из внешней памяти, так и по сети
- Позволяет подгружать платформозависимый код в процессе работы программы
- Позволяет применять к акторам прототипное программирование



# СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ

- Сетевое взаимодействие осуществляется через акторов посредников, которые абстрагируют процесс передачи сообщений через различные сети
- Сетевое взаимодействие может использоваться для миграции акторов между узлами системы
- Также, есть возможность делегирования вычислений отдельным узлам системы
- Вместе с механизмом замены кода, возможно создание систем с изменяемой топологией, добавление функциональности в систему в соответствии с имеющимся оборудованием





**СПАСИБО ЗА ВНИМАНИЕ!**

