Министерство образования и науки Украины  
Харьковский национальный университет радиоэлектроники

Кафедра искусственного интеллекта

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ

по курсу «Мультиагентные системы и технологии»

## Моделирование агентов в Scala

Выполнил:

ст. гр. СШIм-17-3 Филипов И.К

2017

Цель: ознакомление с методами моделирования многоагентных систем на языке Scala с ипользованием пакета scala.actors

ХОД РАБОТЫ

Задачи:

Выполнить становку и настройку scala и sbt.

Выполнить шаги тьюториала [Scala Actors: A Short Tutorial](https://dl.nure.ua/mod/url/view.php?id=23815).

Изучить [Akka toolkit](https://akka.io/). Пройти краткий тьюториал по akka actors: [Simple Scala Akka Actor examples (Hello, world examples)](https://dl.nure.ua/mod/url/view.php?id=23816).

Реализовать решение задачи из вампус-мира на языке Scala с помощью пакета Akka Actors.

<https://github.com/VerglL/scala/tree/master/lab3>

Библиотека Scala Actors предоставляет как асинхронные, так и синхронные сообщения (последние реализованы путем обмена несколькими асинхронными сообщениями). Более того, участники могут общаться с использованием *фьючерсов,* где запросы обрабатываются асинхронно, но возвращают представление , которое позволяет ждать ответа.

Описание библиотеки akka

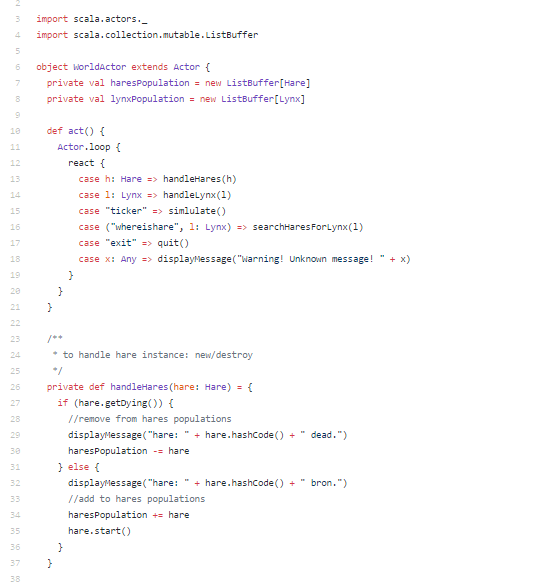
Принцип разработки на akka состоит в том, что все круг акторы. Так же как и ООП исходит из философии, что все круг объекты. Принципиальные отличия же состоят в том, что акторы выполняются параллельно. В то время как ООП код выполняется последовательно и для параллельного исполнения надо делать дополнительные и далеко не всегда простые действия. А так же акторы взаимодействуют между собой не через вызовы методов у объектов, как в ООП, а через отправку сообщений.

Акторы в Akka, описываются как легковесные потоки (green threads). Создание такого актора почти ничего не стоит, их можно создать миллионы. Создатели декларируют, что на 1Gb памяти можно создать порядка 2.5млн акторов. И на одной машине можно достичь скорости обмена порядка 50млн сообщ/сек.

Поэтому код получается слабосвязанным, актору не нужна прямая ссылка другой актор, чтобы отправить ему сообщение. В модели акторов, отсутствует разделяемое состояние. Сообщения приходящие в акторы, обрабатываются последовательно. Получается, что актор не зависит ни от кого. Данные в нем не надо синхронизировать с другими акторами, а код, в отдельно взятом акторе, выполняется «в одном потоке». Ну а как известно, писать однопоточный код гораздо проще чем многопоточный. Но так как у нас акторы выполняются параллельно, то в итоге вся система работает параллельно, равномерно утилизируя все доступное железо. В целом надежность системы получается выше.

У Akka очень много возможностей. Очереди в актёре могут быть реализованы разными способами. Например иметь ограничение по размеру, или сохраняться в БД, или иметь определенную сортировку.Можно управлять потоками, выделяя для какого-то актера отдельный поток или даже пул потоков. Акторы могут работать как в пределах одной машины так и по сети. Есть кластер «из коробки».

Фрагмент кода разработанного на языке Scala в мире Wampus



Выводы в ходе выполнения данной лабораторной работы мы познакомились с функциональным подходом программирования на языке SCALA используя библиотеку akka\_actors. В ходе выполнения были трудности с адаптацией к функциональному подходу программирования, так как он существенно отличается от OOП-стиля.