Министерство образования и науки Украины  
Харьковский национальный университет радиоэлектроники

Кафедра искусственного интеллекта

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ

по курсу «Мультиагентные системы и технологии»

СОЗДАНИЕ КОММУНИКАТИВНЫХ АГЕНТОВ В JADE

Выполнил:

ст. гр. СШIм-17-3 Филипов И.К.

2017

НАЗВАНИЕ РАБОТЫ: создание коммуникативных агентов в Jade.

ЦЕЛЬ РАБОТЫ: познакомить студентов с пакетом Jade.

ХОД РАБОТЫ

* Задача: Выполнить все шаги по построению коммуникативных агентов в Jade, следуя [руководству программиста Jade](http://jade.tilab.com/doc/tutorials/JADEProgramming-Tutorial-for-beginners.pdf).

<https://github.com/VerglL/scala/tree/master/lab1>

JADE является промежуточным слоем программного обеспечения, который предназначен для облегчения разработки мультиагентных систем. Он включает в себя:

* Среду реального времени, в которой агенты JADE могут «существовать». Эта среда должна быть запущена на указанном хосте до того, как первый агент будет запущен.
* Библиотеку классов, которые программист может использовать для разработки их агентов.
* Набор графических инструментов, который позволяет администрировать и отслеживать активность запущенных агентов.

Для создания агента в JADE необходимо просто определить класс, наследующий класс jade.core.Agent и реализовав метод setup():

import  jade.core.Agent;

public class  BookBuyerAgent extends Agent {

protected void setup() {

// Printout a welcome message

    System.out.println(“Hello! Buyer-agent  “+getAID().getName()+” is ready.”);

}

}

После выполнения следующего кода:

C:\jade>java  –classpath <JADE-classes> jade.Boot buyer:BookBuyerAgent

Получен вывод:

..

..

Hello!  Buyer-agent buyer@NBNT2004130496:1099/JADE is ready.

Когда не  предусмотрены никакие действия после печати приветствия, агент всё ещё  запущен. Для того, чтобы завершить его работу нужно использовать метод doDelete(). Подобно методу setup() (который вызывается средой JADE сразу после создания агента, и в который необходимо  включить инициализацию агента), метод takeDown()  вызывается перед завершением работы агента.  Он должен содержать операции по очистке.

Агенты  могут получать аргументы при запуске из командной строки. Эти аргументы могут  быть получены как массив Object, с помощью метода getArguments() класса Agent. Необходимо, чтобы агент BookByuerAgent получал названия книги, которые  необходимо купить через командную строку. Для того, чтобы достичь этого,  модифицируем агент следующим образом:

import  jade.core.Agent;

   import  jade.core.AID;

   public class  BookBuyerAgent extends Agent {

   // The title of the book to buy

   private String targetBookTitle;

   // The list of known seller agents

   private AID[] sellerAgents = {new AID(“seller1”,  AID.ISLOCALNAME),new AID(“seller2”, AID.ISLOCALNAME)};

   // Put agent initializations here

   protected void setup() {

   // Printout a welcome message

   System.out.println(“Hello! Buyer-agent  “+getAID().getName()+” is ready.”);

   // Get the title of the book to buy as a  start-up argument

   Object[] args = getArguments();

   if (args != null && args.length  > 0) {

   targetBookTitle = (String) args[0];

   System.out.println(“Trying to buy  “+targetBookTitle);

   }

   else {

   // Make the agent terminate immediately

   System.out.println(“No book title  specified“);

   doDelete();

   } }

   // Put agent clean-up operations here

   protected void takeDown() {

   // Printout a dismissal message

   System.out.println(“Buyer-agent  “+getAID().getName()+” terminating.”); } }

   Аргументы командной строки заключены  в скобки и разделены пробелами.

C:\jade>java  jade.Boot buyer:BookBuyerAgent(The-Lord-of-the-rings)

   ...

   ...

   5-mag-2008  11.11.00 jade.core.AgentContainerImpl joinPlatform

   INFO:  --------------------------------------

   Agent container  Main-Container@NBNT2004130496 is ready.

   --------------------------------------------

   Hello!  Buyer-agent buyer@NBNT2004130496:1099/JADE is ready.

   Trying to buy  The-Lord-of-the-Rings

Фактическая работа, которую агент должен делать, как правило, осуществляется в рамках «поведения агента». Поведение представляет собой задачу, которую агент может выполнять. Оно реализуется как объект класса, наследующего класс jade.core.behaviours.Behaviour.

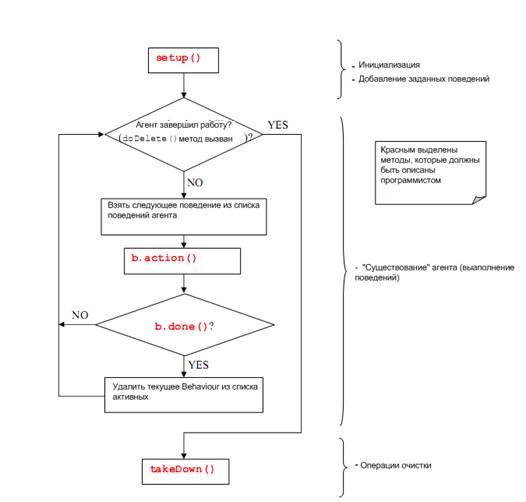
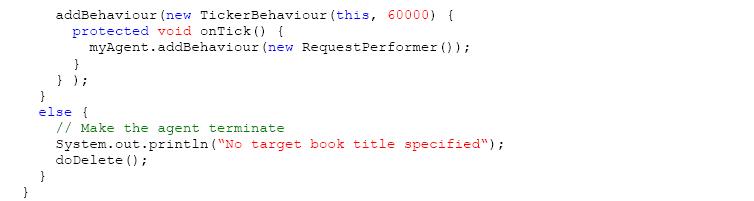


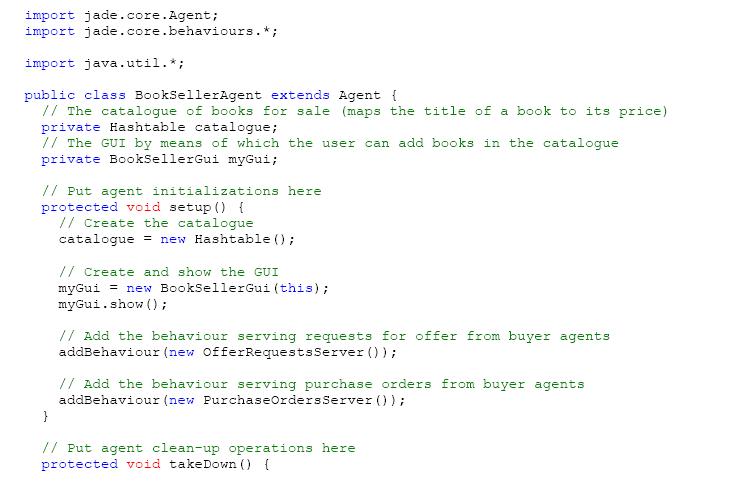
Рисунок 1 - Путь Потока Агента выполнения

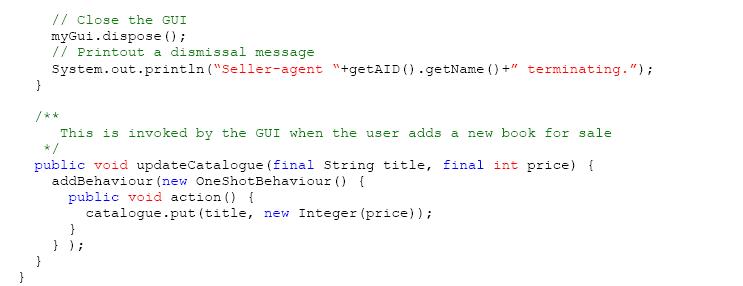
Поведение агента «Книжного покупателя»





Поведения агента «Продавца книг»





Также в ходе работы были реализованы связи агента - асинхронная передача сообщений с помощью класса ACLMESSAGE; реализована блокировка поведения, ждущего сообщения (класс OfferRequestsServer).

ВЫВОД:

В ходе выполнения лабораторной работы было проработано руководство программиста Jade и созданы агенты, продающие книги и агенты, покупающие книги от имени их пользователей (BookBuyerAgent, BookSellerAgent). Также были созданы поведения, система связи между агентами с помощью асинхронных сообщений. Я ознакомился с пакетом Jade.