**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

INTELEKTIKOS PAGRINDAI (P176B101)

**5 laboratorinio darbo ataskaita.**

Atliko:

IFF-4/1 gr. studentas

Mangirdas Kazlauskas

Priėmė:

Doc. Agnė Paulauskaitė-Tarasevičienė

**KAUNAS 2017**

1. Darbo užduotis

5 laboratorinio **darbo tikslas:** Įsidiegti ir susipažinti su JADE programiniu įrankiu.  Susipažinti su agentų elgsenų tipais ir mokėti jas realizuoti.   Darbas susideda iš 4 nuosekliai sunkėjančių dalių. Visų darbų realizacijos yra pateiktos aprašuose.

**Darbo užduotis:** išsiaiškinti ir suprasti, kaip veikia agentinė programa, ir gebėti atlikti įvairias agentų elgsenos modifikacijas gynimo metu.

1. Užduoties sprendimo programos kodas
   1. pirmoji dalis
      1. Agento “Sveikas\_pasauli” realizacija

package Agentai;

import jade.core.Agent;

import jade.core.behaviours.OneShotBehaviour;

import jade.core.behaviours.TickerBehaviour;

import jade.core.behaviours.WakerBehaviour;

import jade.core.behaviours.CyclicBehaviour;

// Arguments: -name Platforma -gui PAG\_Agentas:Agentai.Sveikas\_Pasauli

public class Sveikas\_Pasauli extends Agent {

private int countdown = 10;

private boolean start = false;

private boolean stopAgent = false;

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour(this) {

@Override

public void action() {

System.out.println("Sveikas\_Pasauli");

System.out.println("Mano vardas yra - " + getAID().getLocalName());

}

});

addBehaviour(new TickerBehaviour(this, 1000) {

@Override

protected void onTick() {

if (start){

System.out.println(countdown + " sek");

if (countdown == 0){

stopAgent = true;

}

else{

countdown--;

}

}

}

});

addBehaviour(new WakerBehaviour(this, 3000) {

@Override

protected void onWake(){

start = true;

}

});

addBehaviour(new CyclicBehaviour(this) {

@Override

public void action(){

if(stopAgent){

doDelete();

}

}

});

}

}

### PIRMOSIOS SAVARANKISKOS UŽDUOTIES REALIZACIJA

package Agentai;

import jade.core.Agent;

import jade.core.behaviours.CyclicBehaviour;

import jade.core.behaviours.TickerBehaviour;

import jade.core.behaviours.WakerBehaviour;

// Arguments: -name Platforma -gui SP\_Agentas:Agentai.Sav\_1(Hello)

public class Sav\_1 extends Agent{

private boolean stopAgent = false;

private int counter = 0;

@Override

protected void setup(){

Object[] args = getArguments();

String message = args[0].toString();

//------------------------------------------------------

// Variantas 1

//------------------------------------------------------

// addBehaviour(new TickerBehaviour(this, 1000){

// @Override

// protected void onTick(){

// System.out.println("Current time: " + getTickCount() + " s.");

// }

// });

//

// addBehaviour(new TickerBehaviour(this, 5000){

// @Override

// protected void onTick(){

// System.out.println(message);

// if(getTickCount() == 6){

// doDelete();

// }

// }

// });

//------------------------------------------------------

// Variantas 2

//------------------------------------------------------

// addBehaviour(new TickerBehaviour(this, 5000){

// @Override

// protected void onTick(){

// System.out.println(message);

// counter++;

// if(counter == 6){

// stopAgent = true;

// }

// }

// });

// addBehaviour(new CyclicBehaviour(this) {

// @Override

// public void action(){

// if(stopAgent){

// doDelete();

// }

// }

// });

//------------------------------------------------------

// Variantas 3

//------------------------------------------------------

// addBehaviour(new TickerBehaviour(this, 5000){

// @Override

// protected void onTick(){

// System.out.println(message);

// }

// });

// addBehaviour(new WakerBehaviour(this, 30000){

// @Override

// protected void onWake(){

// doDelete();

// }

// });

//------------------------------------------------------

// Variantas 4

//------------------------------------------------------

addBehaviour(new TickerBehaviour(this, 1000){

@Override

protected void onTick(){

System.out.println("Current time: " + getTickCount() + " s.");

if(getTickCount() % 5 == 0){

System.out.println(message);

}

if(getTickCount() == 30){

doDelete();

}

}

});

}

}

* + 1. **ANTROSIOS SAVARANKIŠKOS UŽDUOTIES REALIZACIJA**

package Agentai;

import jade.core.Agent;

import jade.core.behaviours.TickerBehaviour;

import jade.core.behaviours.WakerBehaviour;

// Arguments: -name Platforma -gui SP\_Agentas:Agentai.Sav\_2(3000)

public class Sav\_2 extends Agent{

@Override

protected void setup(){

Object[] args = getArguments();

int time = Integer.parseInt(args[0].toString());

addBehaviour(new TickerBehaviour(this, 1000){

@Override

protected void onTick(){

System.out.println("Current time: " + getTickCount() + " s.");

}

});

addBehaviour(new WakerBehaviour(this, time) {

@Override

protected void onWake(){

System.out.println("Sveiki aš agentas vardu: " + getAID().getLocalName());

myAgent.addBehaviour(new WakerBehaviour(myAgent, 30000) {

@Override

protected void onWake(){

doDelete();

}

});

}

});

}

}

* + 1. **TREČIOSIOS SAVARANKIŠKOS UŽDUOTIES REALIZACIJA**

package Agentai;

import jade.core.Agent;

import jade.core.behaviours.TickerBehaviour;

import jade.core.behaviours.WakerBehaviour;

// Arguments: -name Platforma -gui SP\_Agentas:Agentai.Sav\_3(3000,Hello,3000)

public class Sav\_3 extends Agent{

@Override

protected void setup(){

Object[] args = getArguments();

int time = Integer.parseInt(args[0].toString());

String message = args[1].toString();

int turnOffAfter = Integer.parseInt(args[2].toString());

addBehaviour(new TickerBehaviour(this, 1000){

@Override

protected void onTick(){

System.out.println("Current time: " + getTickCount() + " s.");

}

});

addBehaviour(new WakerBehaviour(this, time) {

@Override

protected void onWake(){

System.out.println(message);

myAgent.addBehaviour(new WakerBehaviour(myAgent, turnOffAfter) {

@Override

protected void onWake(){

doDelete();

}

});

}

});

}

}

* + 1. KETVIRTOSIOS SAVARANKIŠKOS UŽDUOTIES REALIZACIJA

package Agentai;

import jade.core.Agent;

import jade.core.behaviours.TickerBehaviour;

import jade.core.behaviours.WakerBehaviour;

// Arguments: -name Platforma -gui SP\_Agentas:Agentai.Sav\_4(3000,Hello,3000)

// -name Platforma -gui SP\_Agentas:Agentai.Sav\_4(Hello,3000)

public class Sav\_4 extends Agent{

@Override

protected void setup(){

Object[] args = getArguments();

int time = 5000;

String message;

int turnOffAfter;

if(args.length == 3){

time = Integer.parseInt(args[0].toString());

message = args[1].toString();

turnOffAfter = Integer.parseInt(args[2].toString());

}

else{

message = args[0].toString();

turnOffAfter = Integer.parseInt(args[1].toString());

}

addBehaviour(new TickerBehaviour(this, 1000){

@Override

protected void onTick(){

System.out.println("Current time: " + getTickCount() + " s.");

}

});

addBehaviour(new WakerBehaviour(this, time) {

@Override

protected void onWake(){

System.out.println(message);

myAgent.addBehaviour(new WakerBehaviour(myAgent, turnOffAfter) {

@Override

protected void onWake(){

doDelete();

}

});

}

});

}

}

* 1. **ANTROJI DALIS**
     1. **DEMONSTRACINIŲ AGENTŲ PAGRINDINIS IR PAPILDOMAS REALIZACIJA**

**Agentas „Pagrindinis“:**

package Agentai;

import jade.core.AID;

import jade.core.Agent;

import jade.core.behaviours.CyclicBehaviour;

import jade.wrapper.AgentContainer;

import jade.wrapper.AgentController;

import jade.core.behaviours.OneShotBehaviour;

import jade.lang.acl.ACLMessage;

// Demo

public class Pagrindinis extends Agent{

private AgentController AC = null;

private String pack = "Agentai";

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

System.out.println("----------------------------------------------------");

SukurtiAgenta("Papildomas");

System.out.println("----------------------------------------------------");

}

});

addBehaviour(new CyclicBehaviour() {

@Override

public void action() {

jade.lang.acl.ACLMessage zinute = myAgent.receive();

if(zinute != null){

String vardas;

String turinys;

vardas = zinute.getSender().getName();

vardas = vardas.substring(0, vardas.indexOf("@"));

if (vardas.equals("Papildomas")){

turinys = zinute.getContent();

ACLMessage atsakymas;

atsakymas = new ACLMessage(ACLMessage.INFORM);

atsakymas.addReceiver(new AID(vardas, AID.ISLOCALNAME));

System.out.println("Iš agento " + vardas + " gauta žinutė: ");

System.out.println(turinys);

atsakymas.setContent("Pagrindinis agentas patvirtina gavęs žinutę");

send(atsakymas);

}

}

else{

block();

}

}

});

}

private void SukurtiAgenta(String Pav){

try{

AgentContainer Konteineris = (AgentContainer) getContainerController();

AC = Konteineris.createNewAgent(Pav, pack + "." + Pav, null);

AC.start();

System.out.println("----------------------------------------------------");

System.out.println("Pagrindinis agentas: " + getLocalName());

System.out.println("Pagrindinio agento sukurtas agentas: " + Pav);

System.out.println("Konteineris: " + Konteineris.getContainerName());

System.out.println("----------------------------------------------------");

}

catch(Exception any){

any.printStackTrace();

}

}

}

**Agentas „Papildomas“:**

package Agentai;

import jade.core.AID;

import jade.core.Agent;

import jade.core.behaviours.OneShotBehaviour;

import jade.core.behaviours.TickerBehaviour;

import jade.lang.acl.ACLMessage;

// Demo

public class Papildomas extends Agent{

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

System.out.println("----------------------------------------------------");

System.out.println("Papildomas Agentas");

System.out.println("----------------------------------------------------");

}

});

addBehaviour(new TickerBehaviour(this, 5000) {

@Override

protected void onTick() {

ACLMessage zinute;

zinute = new ACLMessage(ACLMessage.INFORM);

zinute.addReceiver(new AID("PAG\_Agentas", AID.ISLOCALNAME));

zinute.setContent("Papildomo agento žinutės turinys");

send(zinute);

}

});

}

}

* + 1. **PIRMOSIOS SAVARANKIŠKOS UŽDUOTIES REALIZACIJA**

**Agentas “Pagrindinis”:**

// Arguments: -name Platforma -gui PAG\_Agentas:Agentai.Pagrindinis("Zinutes tekstas")

public class Pagrindinis extends Agent{

private AgentController AC = null;

private String pack = "Agentai";

@Override

protected void setup(){

Object args[] = getArguments();

String message = args[0].toString();

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

System.out.println("Pagrindinis agentas pradėjo darbą.");

SukurtiAgenta("Papildomas");

}

});

addBehaviour(new TickerBehaviour(this, 2000) {

@Override

protected void onTick() {

ACLMessage zinute;

zinute = new ACLMessage(ACLMessage.INFORM);

zinute.addReceiver(new AID("Papildomas", AID.ISLOCALNAME));

zinute.setContent(message);

send(zinute);

}

});

// Testavimui, ar papildomas agentas prieš darbo pabaigą išsiunčia žinutę

// addBehaviour(new CyclicBehaviour() {

// @Override

// public void action() {

// jade.lang.acl.ACLMessage zinute = myAgent.receive();

// if(zinute != null){

// String vardas;

// vardas = zinute.getSender().getName();

// vardas = vardas.substring(0, vardas.indexOf("@"));

// if (vardas.equals("Papildomas")){

// System.out.println("OK");

// }

// }

// else{

// block();

// }

// }

// });

}

private void SukurtiAgenta(String Pav){

try{

AgentContainer Konteineris = (AgentContainer) getContainerController();

System.out.println("Sukuriamas papildomas agentas.");

AC = Konteineris.createNewAgent(Pav, pack + "." + Pav, null);

AC.start();

}

catch(Exception any){

any.printStackTrace();

}

}

}

**Agentas “Papildomas”:**

public class Papildomas extends Agent{

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

System.out.println("Papildomas agentas pradėjo darbą.");

}

});

addBehaviour(new CyclicBehaviour() {

@Override

public void action() {

jade.lang.acl.ACLMessage zinute = myAgent.receive();

if(zinute != null){

String vardas;

String turinys;

vardas = zinute.getSender().getName();

vardas = vardas.substring(0, vardas.indexOf("@"));

if (vardas.equals("PAG\_Agentas")){

turinys = zinute.getContent();

System.out.println(turinys);

ACLMessage confirm = new ACLMessage(ACLMessage.CONFIRM);

confirm.addReceiver(new AID("PAG\_Agentas", AID.ISLOCALNAME));

send(confirm);

}

}

else{

block();

}

}

});

}

}

* + 1. **ANTROSIOS SAVARANKIŠKOS UŽDUOTIES REALIZACIJA**

**Agentas “Pagrindinis”:**

// Arguments: -name Platforma -gui PAG\_Agentas:Agentai.Pagrindinis(5000)

public class Pagrindinis extends Agent{

private AgentController AC = null;

private String pack = "Agentai";

@Override

protected void setup(){

Object args[] = getArguments();

String turnOffTime = args[0].toString();

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

System.out.println("Pagrindinis agentas pradėjo darbą.");

SukurtiAgenta("Papildomas");

}

});

addBehaviour(new TickerBehaviour(this, 5000) {

@Override

protected void onTick() {

ACLMessage zinute;

zinute = new ACLMessage(ACLMessage.INFORM);

zinute.addReceiver(new AID("Papildomas", AID.ISLOCALNAME));

zinute.setContent(turnOffTime);

send(zinute);

}

});

}

private void SukurtiAgenta(String Pav){

try{

AgentContainer Konteineris = (AgentContainer) getContainerController();

System.out.println("Sukuriamas papildomas agentas.");

AC = Konteineris.createNewAgent(Pav, pack + "." + Pav, null);

AC.start();

}

catch(Exception any){

any.printStackTrace();

}

}

}

**Agentas “Papildomas”:**

public class Papildomas extends Agent{

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

System.out.println("Papildomas agentas pradėjo darbą.");

}

});

addBehaviour(new CyclicBehaviour() {

@Override

public void action() {

jade.lang.acl.ACLMessage zinute = myAgent.receive();

if(zinute != null){

String vardas;

String turnOffAfter;

vardas = zinute.getSender().getName();

vardas = vardas.substring(0, vardas.indexOf("@"));

if (vardas.equals("PAG\_Agentas")){

turnOffAfter = zinute.getContent();

System.out.println(turnOffAfter);

ACLMessage confirm = new ACLMessage(ACLMessage.CONFIRM);

confirm.addReceiver(new AID("PAG\_Agentas", AID.ISLOCALNAME));

send(confirm);

int time = Integer.parseInt(turnOffAfter);

myAgent.addBehaviour(new WakerBehaviour(myAgent, time){

@Override

protected void onWake(){

doDelete();

ACLMessage zinute;

// Pagal užduotį reikia siųsti Confirm tipo žinutę

// zinute = new ACLMessage(ACLMessage.INFORM);

zinute = new ACLMessage(ACLMessage.INFORM);

zinute.addReceiver(new AID("PAG\_Agentas", AID.ISLOCALNAME));

zinute.setContent("Papildomas agentas baigia darbą");

send(zinute);

}

});

}

}

else{

block();

}

}

});

}

}

* + 1. **TREČIOSIOS SAVARANKISKOS UŽDUOTIES REALIZACIJA**

**Agentas “Pagrindinis”:**

// Arguments: -name Platforma -gui PAG\_Agentas:Agentai.Pagrindinis("Zinutes tekstas",5000)

public class Pagrindinis extends Agent{

private final String pack = "Agentai";

final static String name\_Agent = "Papildomas";

final static String name\_Agent2 = "Papildomas2";

private AgentController AC;

private AgentController AC2;

private boolean switcher = true; // true - Papildomas, false - papildomas 2

@Override

protected void setup(){

Object args[] = getArguments();

String turnOffTime = args[1].toString();

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

System.out.println("Pagrindinis agentas pradėjo darbą.");

AC = SukurtiAgenta(name\_Agent, args);

AC2 = SukurtiAgenta(name\_Agent2, args);

}

});

addBehaviour(new TickerBehaviour(this, 5000) {

@Override

protected void onTick() {

ACLMessage zinute;

String agentName = switcher ? name\_Agent : name\_Agent2;

zinute = new ACLMessage(ACLMessage.INFORM);

zinute.addReceiver(new AID(agentName, AID.ISLOCALNAME));

zinute.setContent(turnOffTime);

send(zinute);

}

});

addBehaviour(new CyclicBehaviour() {

@Override

public void action() {

jade.lang.acl.ACLMessage zinute = myAgent.receive();

if(zinute != null){

String vardas;

vardas = zinute.getSender().getName();

vardas = vardas.substring(0, vardas.indexOf("@"));

if (vardas.equals(name\_Agent)){

switcher = false;

try {

System.out.println(zinute.getContent());

AC2.activate();

} catch (StaleProxyException ex) {

Logger.getLogger(Pagrindinis.class.getName()).log(Level.SEVERE, null, ex);

}

}

else if(vardas.equals(name\_Agent2)){

switcher = true;

try {

System.out.println(zinute.getContent());

AC.activate();

} catch (StaleProxyException ex) {

Logger.getLogger(Pagrindinis.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

else{

block();

}

}

});

}

private AgentController SukurtiAgenta(String Pav, Object[] args){

AgentController AController = null;

try{

AgentContainer Konteineris = (AgentContainer) getContainerController();

System.out.println("Sukuriamas agentas: " + Pav);

AController = Konteineris.createNewAgent(Pav, pack + "." + name\_Agent, args);

AController.start();

}

catch(Exception any){

any.printStackTrace();

}

return AController;

}

}

**Agentas “Papildomas”:**

public class Papildomas extends Agent{

@Override

protected void setup(){

Object args[] = getArguments();

String arg\_message = args[0].toString();

addBehaviour(new OneShotBehaviour(this) {

@Override

public void action() {

System.out.println(getAID().getLocalName() + " agentas pradėjo darbą.");

if (getAID().getLocalName().equals(Pagrindinis.name\_Agent2)){

doSuspend();

}

}

});

addBehaviour(new CyclicBehaviour(this) {

@Override

public void action() {

jade.lang.acl.ACLMessage zinute = myAgent.receive();

if(zinute != null){

String vardas;

String turinys;

vardas = zinute.getSender().getName();

vardas = vardas.substring(0, vardas.indexOf("@"));

if (vardas.equals("PAG\_Agentas")){

ACLMessage informAboutMessage = new ACLMessage(ACLMessage.INFORM);

informAboutMessage.addReceiver(new AID("PAG\_Agentas", AID.ISLOCALNAME));

//send(informAboutMessage);

int time = Integer.parseInt(zinute.getContent());

System.out.println(getAID().getLocalName() + ": " + time);

myAgent.addBehaviour(new WakerBehaviour(myAgent, time){

@Override

protected void onWake(){

ACLMessage confirm = new ACLMessage(ACLMessage.CONFIRM);

confirm.addReceiver(new AID("PAG\_Agentas", AID.ISLOCALNAME));

confirm.setContent(getAID().getLocalName() + ": baigiau darbą");

System.out.println(getAID().getLocalName() + ": " + arg\_message);

send(confirm);

doSuspend();

}

});

}

}

else{

block();

}

}

});

}

}

* 1. **TREČIOJI DALIS**

**Agentas “Pagrindinis”:**

// Arguments: -name Platforma -gui PAG\_Agentas:Agentai.Pagrindinis(Turinys)

public class Pagrindinis extends Agent{

private final String pack = "Agentai";

final static String name\_Agent = "Papildomas";

private AgentController AC;

@Override

protected void setup(){

Object[] args = getArguments();

String Argumentas = args[0].toString();

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

String vardas;

vardas = "Klase";

System.out.println("Pagrindinis agentas pradėjo darbą.");

ContainerController cc = SukurtiKonteineri(myAgent.getProperty(Profile.MAIN\_HOST, null),

myAgent.getProperty(Profile.MAIN\_PORT, null),

vardas);

SukurtiAgentaKonteineryje(name\_Agent, cc);

}

});

addBehaviour(new TickerBehaviour(this, 2000) {

@Override

protected void onTick() {

ACLMessage cfp = new ACLMessage(ACLMessage.PROPOSE);

cfp.addReceiver(new AID(name\_Agent, AID.ISLOCALNAME));

cfp.setContent(Argumentas);

myAgent.send(cfp);

}

});

}

private ContainerController SukurtiKonteineri(String Hostas, String Portas, String Vardas){

Runtime rt = Runtime.instance();

Profile p = new ProfileImpl();

p.setParameter(Profile.MAIN\_HOST, Hostas);

p.setParameter(Profile.MAIN\_PORT, Portas);

p.setParameter(Profile.CONTAINER\_NAME, Vardas);

return rt.createAgentContainer(p);

}

private void SukurtiAgentaKonteineryje(String Pav, ContainerController Konteineris){

try{

AC = Konteineris.createNewAgent(Pav, pack + "." + Pav, null);

System.out.println("Pagrindinis agentas: " + getLocalName());

System.out.println("Sukuriamas agentas: " + Pav);

System.out.println("Konteinerio pavadinimas: " + Konteineris.getContainerName());

AC.start();

}

catch(Exception any){

any.printStackTrace();

}

}

}

**Agentas “Papildomas”:**

public class Papildomas extends Agent{

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour(this) {

@Override

public void action() {

System.out.println(getAID().getLocalName() + " agentas pradėjo darbą.");

}

});

addBehaviour(new CyclicBehaviour(this) {

@Override

public void action() {

jade.lang.acl.ACLMessage zinute = myAgent.receive();

if(zinute != null){

String vardas;

vardas = zinute.getSender().getName();

vardas = vardas.substring(0, vardas.indexOf("@"));

if (vardas.equals("PAG\_Agentas")){

Random rnd = new Random();

boolean accept = rnd.nextBoolean();

if(accept){

System.out.println("Pasiūlymas priimamas.");

ACLMessage message = new ACLMessage(ACLMessage.ACCEPT\_PROPOSAL);

message.addReceiver(new AID(vardas, AID.ISLOCALNAME));

message.setContent("Atmetu pasiūlymą.");

send(message);

}

else{

System.out.println("Pasiūlymas atmetamas.");

ACLMessage message = new ACLMessage(ACLMessage.REJECT\_PROPOSAL);

message.addReceiver(new AID(vardas, AID.ISLOCALNAME));

message.setContent("Priimu pasiūlymą.");

send(message);

}

}

}

else{

block();

}

}

});

}

}

* 1. **KETVIRTOJI DALIS**

**Agentas “Pagrindinis”:**

// Arguments: -name Platforma -gui PAG\_Agentas:Agentai.Pagrindinis

public class Pagrindinis extends Agent{

private final String pack = "Agentai";

final static String name\_Agent = "Papildomas";

final static String name\_Agent2 = "Papildomas2";

final static String name\_Agent3 = "Mobilus";

private AgentController AC;

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

String vardas = "Klase";

String vardas2 = "Klase2";

System.out.println("Pagrindinis agentas pradėjo darbą.");

ContainerController cc = SukurtiKonteineri(myAgent.getProperty(Profile.MAIN\_HOST, null),

myAgent.getProperty(Profile.MAIN\_PORT, null),

vardas);

ContainerController cr = SukurtiKonteineri(myAgent.getProperty(Profile.MAIN\_HOST, null),

myAgent.getProperty(Profile.MAIN\_PORT, null),

vardas2);

SukurtiAgentaKonteineryje(name\_Agent, cc);

SukurtiAgentaKonteineryje(name\_Agent2, cr);

SukurtiAgenta(name\_Agent3);

}

});

}

private ContainerController SukurtiKonteineri(String Hostas, String Portas, String Vardas){

Runtime rt = Runtime.instance();

Profile p = new ProfileImpl();

p.setParameter(Profile.MAIN\_HOST, Hostas);

p.setParameter(Profile.MAIN\_PORT, Portas);

p.setParameter(Profile.CONTAINER\_NAME, Vardas);

return rt.createAgentContainer(p);

}

private void SukurtiAgentaKonteineryje(String Pav, ContainerController Konteineris){

try{

AC = Konteineris.createNewAgent(Pav, pack + "." + Pav, null);

System.out.println("Pagrindinis agentas: " + getLocalName());

System.out.println("Sukuriamas agentas: " + Pav);

System.out.println("Konteinerio pavadinimas: " + Konteineris.getContainerName());

AC.start();

}

catch(Exception any){

any.printStackTrace();

}

}

private void SukurtiAgenta(String Pav){

try{

AgentContainer Konteineris = (AgentContainer) getContainerController();

System.out.println("Sukuriamas papildomas agentas.");

AC = Konteineris.createNewAgent(Pav, pack + "." + Pav, null);

AC.start();

}

catch(Exception any){

any.printStackTrace();

}

}

}

**Agentas “Papildomas”:**

public class Papildomas extends Agent{

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour(this) {

@Override

public void action() {

System.out.println(getAID().getLocalName() + " agentas pradėjo darbą.");

}

});

addBehaviour(new CyclicBehaviour(this) {

@Override

public void action() {

jade.lang.acl.ACLMessage zinute = myAgent.receive();

if(zinute != null){

String vardas;

vardas = zinute.getSender().getName();

vardas = vardas.substring(0, vardas.indexOf("@"));

if (vardas.equals("Mobilus")){

Random rnd = new Random();

boolean accept = rnd.nextBoolean();

if(accept){

System.out.println(getAID().getLocalName() + ": Pasiūlymas priimamas.");

ACLMessage message = new ACLMessage(ACLMessage.ACCEPT\_PROPOSAL);

message.addReceiver(new AID(vardas, AID.ISLOCALNAME));

message.setContent("Accept");

send(message);

}

else{

System.out.println(getAID().getLocalName() + ": Pasiūlymas atmetamas.");

ACLMessage message = new ACLMessage(ACLMessage.REJECT\_PROPOSAL);

message.addReceiver(new AID(vardas, AID.ISLOCALNAME));

message.setContent("Reject");

send(message);

}

}

}

else{

block();

}

}

});

}

}

**Agentas “Papildomas2”:**

public class Papildomas2 extends Agent{

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour(this) {

@Override

public void action() {

System.out.println(getAID().getLocalName() + " agentas pradėjo darbą.");

}

});

addBehaviour(new CyclicBehaviour(this) {

@Override

public void action() {

jade.lang.acl.ACLMessage zinute = myAgent.receive();

if(zinute != null){

String vardas;

vardas = zinute.getSender().getName();

vardas = vardas.substring(0, vardas.indexOf("@"));

if (vardas.equals("Mobilus")){

Random rnd = new Random();

boolean accept = rnd.nextBoolean();

if(accept){

System.out.println(getAID().getLocalName() + ": Pasiūlymas priimamas.");

ACLMessage message = new ACLMessage(ACLMessage.ACCEPT\_PROPOSAL);

message.addReceiver(new AID(vardas, AID.ISLOCALNAME));

message.setContent("Accept");

send(message);

}

else{

System.out.println(getAID().getLocalName() + ": Pasiūlymas atmetamas.");

ACLMessage message = new ACLMessage(ACLMessage.REJECT\_PROPOSAL);

message.addReceiver(new AID(vardas, AID.ISLOCALNAME));

message.setContent("Reject");

send(message);

}

}

}

else{

block();

}

}

});

}

}

**Agentas “Mobilus”:**

public class Mobilus extends Agent{

private int papildomas\_accepted = 0;

private int papildomas\_all = 0;

private int papildomas2\_accepted = 0;

private int papildomas2\_all = 0;

private boolean switcher = true; // true - Klase, false - Klase2

@Override

protected void setup(){

addBehaviour(new OneShotBehaviour(this) {

@Override

public void action() {

System.out.println(getAID().getLocalName() + " agentas pradėjo darbą.");

}

});

addBehaviour(new TickerBehaviour(this, 1000) {

@Override

protected void onTick() {

if(getTickCount() == 21){

System.out.println("----------------------------------------------------");

System.out.println("Mobilusis agentas atliko 10 iteracijų, rezultatai:");

System.out.println("Agentas Papildomas: priimti - " + papildomas\_accepted + ", atmesti - " + (papildomas\_all - papildomas\_accepted) + ", viso - " + papildomas\_all + ".");

System.out.println("Agentas Papildomas2: priimti - " + papildomas2\_accepted + ", atmesti - " + (papildomas2\_all - papildomas2\_accepted) + ", viso - " + papildomas2\_all + ".");

System.out.println("----------------------------------------------------");

}

else if(getTickCount() <= 20){

ContainerID dest = new ContainerID();

String name = switcher ? "Klase" : "Klase2";

dest.setName(name);

myAgent.doMove(dest);

}

}

});

addBehaviour(new CyclicBehaviour(this) {

@Override

public void action() {

jade.lang.acl.ACLMessage zinute = myAgent.receive();

if(zinute != null){

String vardas;

vardas = zinute.getSender().getName();

vardas = vardas.substring(0, vardas.indexOf("@"));

if (vardas.equals("Papildomas")){

String content = zinute.getContent();

papildomas\_accepted += content.equals("Accept") ? 1 : 0;

papildomas\_all++;

}

else if(vardas.equals("Papildomas2")){

String content = zinute.getContent();

papildomas2\_accepted += content.equals("Accept") ? 1 : 0;

papildomas2\_all++;

}

}

else{

block();

}

}

});

}

@Override

protected void afterMove(){

addBehaviour(new OneShotBehaviour() {

@Override

public void action() {

String containerName = myAgent.getProperty(Profile.CONTAINER\_NAME, null);

if (containerName == null){

containerName = "Main-Container";

}

System.out.println("----------------------------------------------------");

System.out.println(getAID().getLocalName() + " permigravo į " + containerName);

if(containerName.equals("Klase") || containerName.equals("Klase2")){

ACLMessage message = new ACLMessage(ACLMessage.PROPOSE);

String receiver = containerName.equals("Klase") ? "Papildomas" : "Papildomas2";

message.addReceiver(new AID(receiver, AID.ISLOCALNAME));

message.setContent("Propose message.");

myAgent.send(message);

switcher = !switcher;

}

}

});

}

}