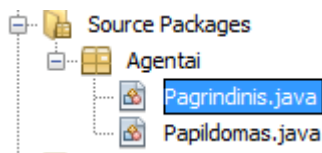


Ketvirtas laboratorinis darbas Nr.4

Darbo tikslas:

- Sukurti mobilųjį agentą, kuris sugebėtų migruoti tarp skirtingų konteinerių. Mobilusis agentas taip pat turi surinkti informaciją iš kituose konteineriuose esančių agentų.
- Po 10 iteracijų išvedamas atmetų ir priimtų užklausų santykis.

1. Modifikuoti bei papildyti trečiajame laboratoriniame darbe sukurtus agentus.



2. Pagrindinio agento papildymas

2.1. Papildykite programos kodą dalyje (**OneShotBehaviour**)

```
String Vardas, Vardas2;
Vardas = "Klase";
Vardas2 = "Klase2";
```

2.2. Pašaliname arba užkomentuojame **TickerBehaviour**, šios dalies vykdomajame etape neprireiks.

2.3. Sukuriam 2 skirtingus konteinerius ir juose agentus su vardais **Papildomas** ir **Papildomas2**. Taip pat pagrindiniame konteineryje sukuriame agentą **Mobilus**.

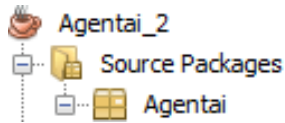
```
ContainerController cc = SukurtiKonteineri
(myAgent.getProperty(Profile.MAIN_HOST, null),
 myAgent.getProperty(Profile.MAIN_HOST, null), Vardas);
ContainerController cr = SukurtiKonteineri
(myAgent.getProperty(Profile.MAIN_HOST, null),
 myAgent.getProperty(Profile.MAIN_HOST, null), Vardas2);
SukurtiAgentąKonteineryje("Papildomas2", cr);
SukurtiAgentąKonteineryje("Papildomas", cc);
SukurtiAgentą("Mobilus");
```

3. Papildykite „Papildomas“ agentą

3.1. Papildykite **Papildomas** agento kodą taip, kad gavus žinutę iš agento **Mobilus** su 50% tikimybe atmetų arba patvirtintų užklausą.

4. Sukurkite „Papildomas2“ agentą

4.1. Pakete „Agentai“ sukurkite naują **Papildomas2** klasę.



4.2. **Papildomas2** agento kodas yra identiškas **Papildomas** agento kodui.

5. Sukurkite „Mobilus“ agentą

Šis mobilus agentas migruos per konteinerius „**Klase**“ ir „**Klase2**“ ir iš agentų „**Papildomas**“ ir „**Papildomas2**“ gaus atsakymą į pateiktas užklausa. Agentas kaups informaciją apie patvirtintas arba atmestas užklausas ir po 10 iteracijų išves pranešimą informaciniame lange (pranešimas įtraukia patvirtintų užklausių skaičių; atmestų užklausių skaičius; bendrą užklausių skaičių;)

5.1. Prie agento vienkartinės elgsenos (**TickerBehaviour**) pridėsime veiksmus, kurie perkels agentą į kitą konteinerį **Klase**.

```
String containerName = "Klase";
ContainerID destination = new ContainerID();
destination.setName(containerName);
myAgent.doMove(destination);
```

Naudojamų funkcijų paaiškinimas:

- **destination.setName(String name)** – konteinerio, į kurį norima perkelti agentą, vardo name nustatymas.
 - **agent.doMove(destination)** – komanda agentui migruoti į tikslo vietos konteinerį.
- 5.2. Papildikyte **TickerBehaviour** elgseną taip, kad kas 1s perkeltų agentą į **Klase**, o po to į **Klase2**. Po dešimties iteracijų išvestų informaciniame lange pranešimą, įtraukiantį patvirtintų, atmestų ir viso užklausių skaičių.
- 5.3. Pridėkite agentui **CyclicBehaviour** elgseną. Parašykite programos kodą, kad gavus žinutes iš agentų **Papildomas** ir **Papildomas2** skaičiuotų, atmestų, patvirtintų pranešimų kiekį.
- 5.4. Papildykite programos kodą po **setup()** dalies

```

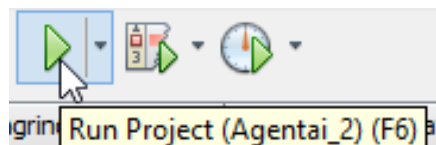
protected void afterMove()
{
    addBehaviour(new OneShotBehaviour(this)
    {
        @Override
        public void action()
        {
            String name = myAgent.getProperty(Profile.CONTAINER_NAME, null);
            if (name == null)
            {
                name = "Main-Container";
            }
            System.out.println(getAID().getLocalName() + " moved to " + name);
            if (name.equals("Klase"))
            {
                //Programos kodas
            }
            if (name.equals("Klase2"))
            {
                //Programos kodas
            }
        }
    });
}

```

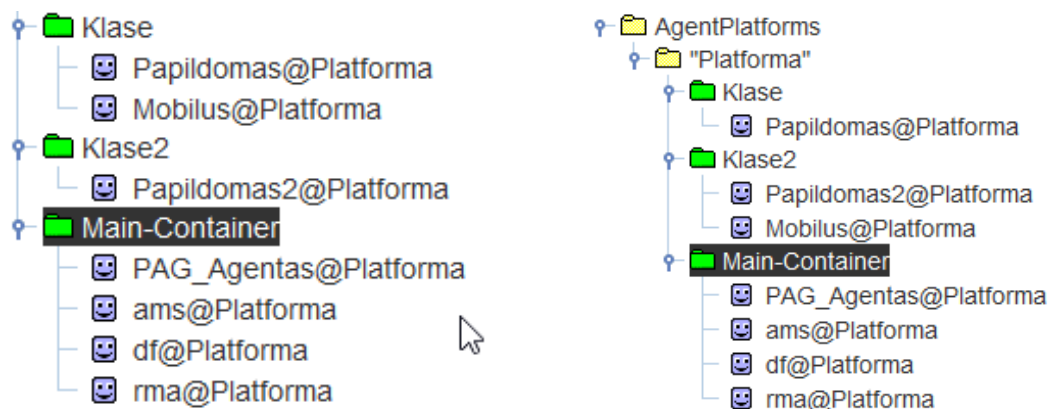
Agento elgsena, kuri yra atliekama po migracijos į kitą konteinerį, yra aprašoma metode **afterMove()**.

5.5. Papildykite programos kodą taip, kad atvykęs į konteinerį mobilus agentas išsiųstų prašymą jame esančiui agentui.

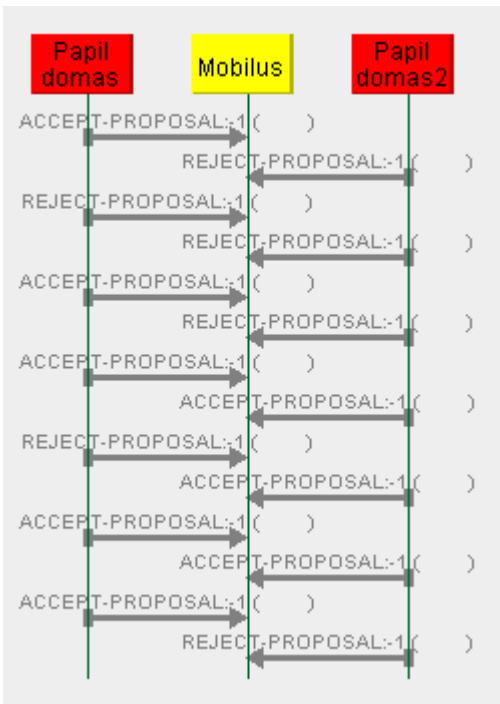
5.6. Paledžiame programą.



5.7. Matome kaip Mobilus agentas peršoka iš vieno konteinerio į kitą.



5.8. Komunikavimas tarp agentų.



5.9. Informacinis langas po 10 iteracijų.

```
-----
Papildomas agentas
prieme: 5 | Atmete:5 | Viso: 10
-----
Papildomas2 agentas
prieme: 4 | Atmete:6 | Viso: 10
-----
```

Punktų realizacijos

3.1 Punkto realizacija:

```
if (vardas.equals("Mobilus"))
{
    turinys = zinute.getContent();
    ACLMessage atsakymas;
    int randomNum = 0 + (int) (Math.random() * 2);
    String ats = Integer.toString(randomNum);
    switch (randomNum)
    {
        case 0:
            atsakymas = new ACLMessage(ACLMessage.REJECT_PROPOSAL);
            atsakymas.addReceiver(new AID(vardas, AID.ISLOCALNAME));

            System.out.println("Papildomas2 agentas atmeta pasiulima");
            atsakymas.setContent("atmeta");
            send(atsakymas);
            break;

        case 1:
            atsakymas = new ACLMessage(ACLMessage.ACCEPT_PROPOSAL);
            atsakymas.addReceiver(new AID(vardas, AID.ISLOCALNAME));

            System.out.println("Papildomas2 agentas priima pasiulima");
            atsakymas.setContent("priima");
            send(atsakymas);
            break;
    }
}
```

3.4 Punkto realizacija:

```
if(zinute != null)
{
    vardas = zinute.getSender().getName();
    vardas = vardas.substring(0, vardas.indexOf("@"));
    if (vardas.equals("Mobilus"))
    {
        turinys = zinute.getContent();
        ACLMessage atsakymas;
        int randomNum = 0 + (int) (Math.random() * 2);
        String ats = Integer.toString(randomNum);
        switch (randomNum)
        {
            case 0:
                atsakymas = new ACLMessage(ACLMessage.REJECT_PROPOSAL);
                atsakymas.addReceiver(new AID(vardas, AID.ISLOCALNAME));

                System.out.println("Papildomas2 agentas atmeta pasiulima");
                atsakymas.setContent("atmeta");
                send(atsakymas);
                break;

            case 1:
                atsakymas = new ACLMessage(ACLMessage.ACCEPT_PROPOSAL);
                atsakymas.addReceiver(new AID(vardas, AID.ISLOCALNAME));

                System.out.println("Papildomas2 agentas priima pasiulima");
                atsakymas.setContent("priima");
                send(atsakymas);
                break;
        }
    }
}
```

5.2 Punkto realizacija:

```
addBehaviour(new TickerBehaviour(this, 1000)
{
    protected void onTick()
    {
        if(iter >= 10)
        {
            System.out.println("-----");
            System.out.println("Papildomas agentas");
            System.out.println("prieme: "+ pb+" | Atmete:"+ (pv-pb)+" | Viso: "+ pv);
            System.out.println("-----");
            System.out.println("Papildomas2 agentas");
            System.out.println("prieme: "+ pb2+" | Atmete:"+ (pv2-pb2)+" | Viso: "+ pv2);
            System.out.println("-----");
            iter = 0;
        }
        if(!k)
        {
            String containerName = "Klase";
            ContainerID destination = new ContainerID();
            destination.setName(containerName);
            myAgent.doMove(destination);
        }
        if(k)
        {
            String containerName2 = "Klase2";
            ContainerID destination2 = new ContainerID();
            destination2.setName(containerName2);
            myAgent.doMove(destination2);
            iter++;
        }
    }
});
```

5.3 Punkto realizacija:

```
addBehaviour(new CyclicBehaviour(this)
{
    @Override
    public void action() {
        String name;
        String content;
        ACLMessage msg = myAgent.receive();
        if(msg != null)
        {
            name = msg.getSender().getName();
            name = name.substring(0, name.indexOf("@"));
            if(name.equals("Papildomas"))
            {
                content = msg.getContent();
                if(content.equals("atmeta")) {pv++;}
                if(content.equals("priima")) {pb++;pv++;}
            }
            if(name.equals("Papildomas2"))
            {
                content = msg.getContent();
                if(content.equals("atmeta")) {pv2++;}
                if(content.equals("priima")) {pb2++;pv2++;}
            }
        }
        else
        {
            block();
        }
    }
});
```

5.5 Punkto realizacija:

```
if (name.equals("Klase"))
{
    ACLMessage msg;
    msg = new ACLMessage(ACLMessage.PROPOSE);
    msg.addReceiver(new AID("papildomas", AID.ISLOCALNAME));
    msg.setContent("Uzklausa");
    send(msg);
    k = true;
}
if (name.equals("Klase2"))
{
    ACLMessage msg;
    msg = new ACLMessage(ACLMessage.PROPOSE);
    msg.addReceiver(new AID("papildomas2", AID.ISLOCALNAME));
    msg.setContent("Uzklausa");
    send(msg);
    k = false;
}
```