

Pirmasis laboratorinis darbas Nr. 1

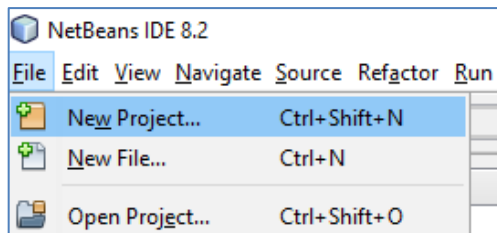
Darbo tikslas:

- Įsidiegti ir susipažinti su JADE programiniu įrankiu.
- Realizuoti pirmąjį standartinį “Sveikas_Pasauli” agentą.
- Susipažinti su įvairiais agentų elgsenų tipais ir mokėti juos realizuoti.

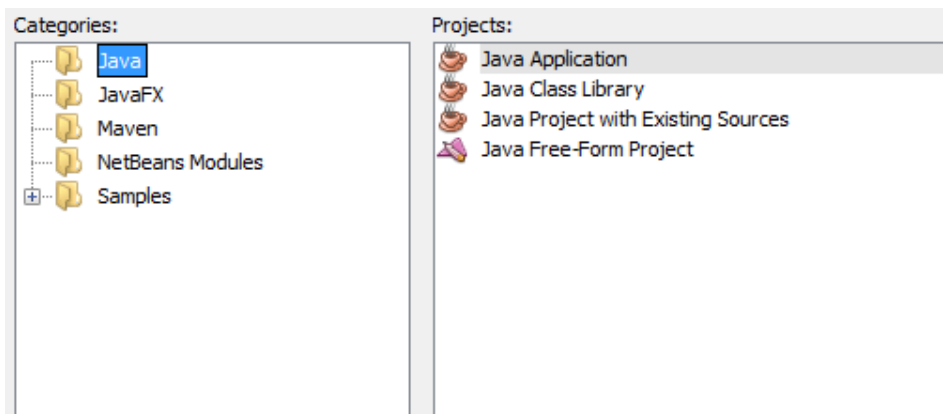
1. NetBeans programinių įrankių įdiegimas ir projekto sukūrimas

1.1. Parsisiųskite NetBeans IDE (<http://netbeans.org/downloads/>)

1.2. Atsidarykite NetBeans ir susikurkite naują projektą: **File -> New Project**

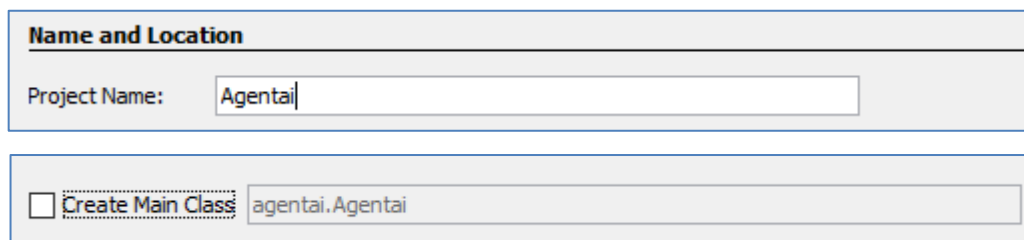


1.3. Pasirinkite **Java-> Java Application**. Spauskite **Next >**

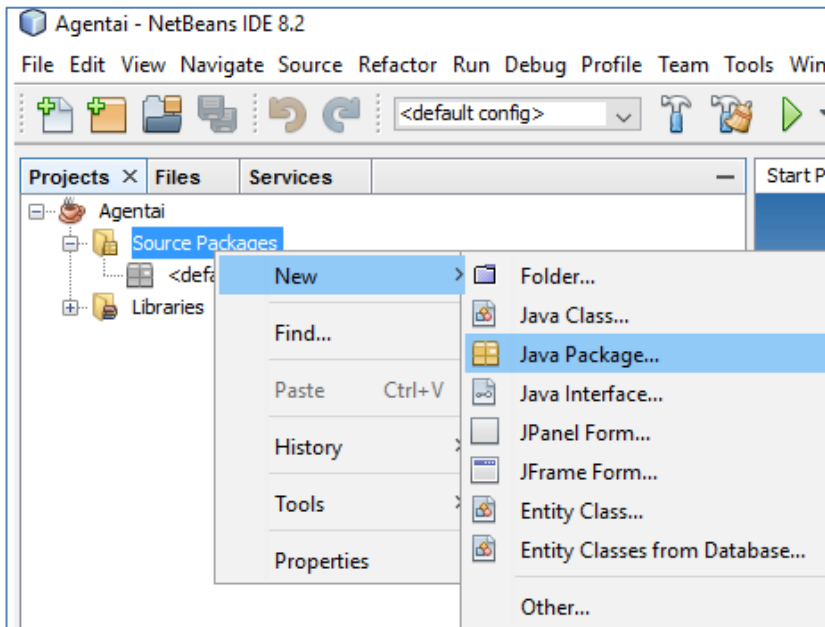


1.4. Įrašykite projekto pavadinimą:

Project Name: Agentai. Nuimkite varnelę **Create Main Class** ir spauskite **Finish**.



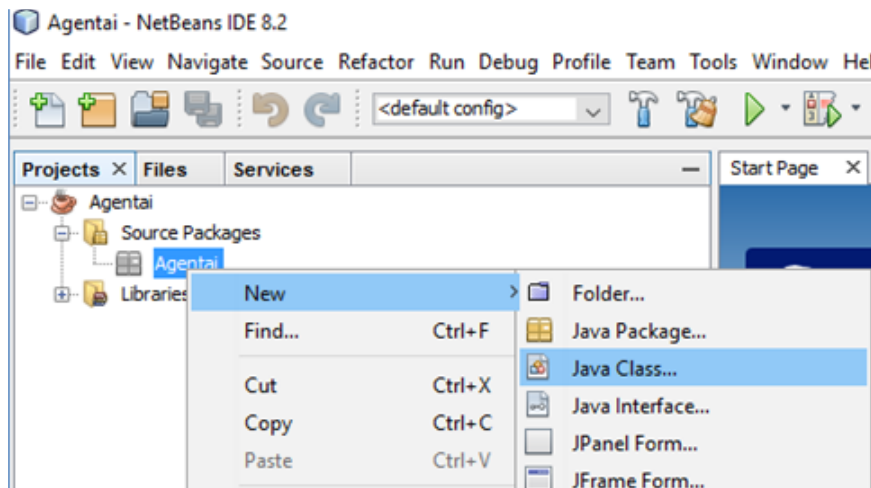
- 1.5. Sukurkite naują paketą. Dešiniuoju pelės mygtuku įvykdysite komandų eilutę:
Source Packages -> New -> Java Package



- 1.6. Įrašykite paketo pavadinimą: **Package name: Agentai**. Spauskite **Finish**.

Name and Location	
Package Name:	Agentai

- 1.7. Ant sukurto paketo spauskite dešinį pelės mygtuką ir pasirinkite:
agentai -> New -> Java Class



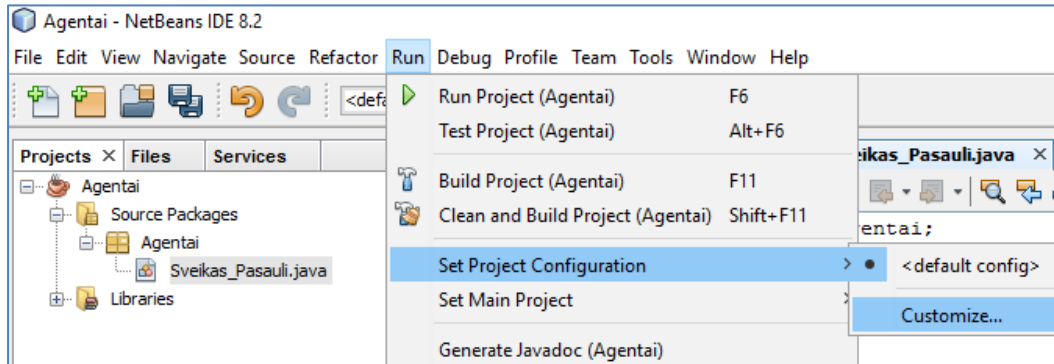
- 1.8. Įrašykite klasės pavadinimą: **Class Name: Sveikas_Pasauli** ir spauskite **Finish**.

Name and Location	
Class Name:	Sveikas_Pasauli

2. JADE programinių įrankių įdiegimas ir bibliotekos įtraukimas

2.1. Parsisiųskite JADE programinius įrankius (<http://jade.tilab.com/download/jade/>).

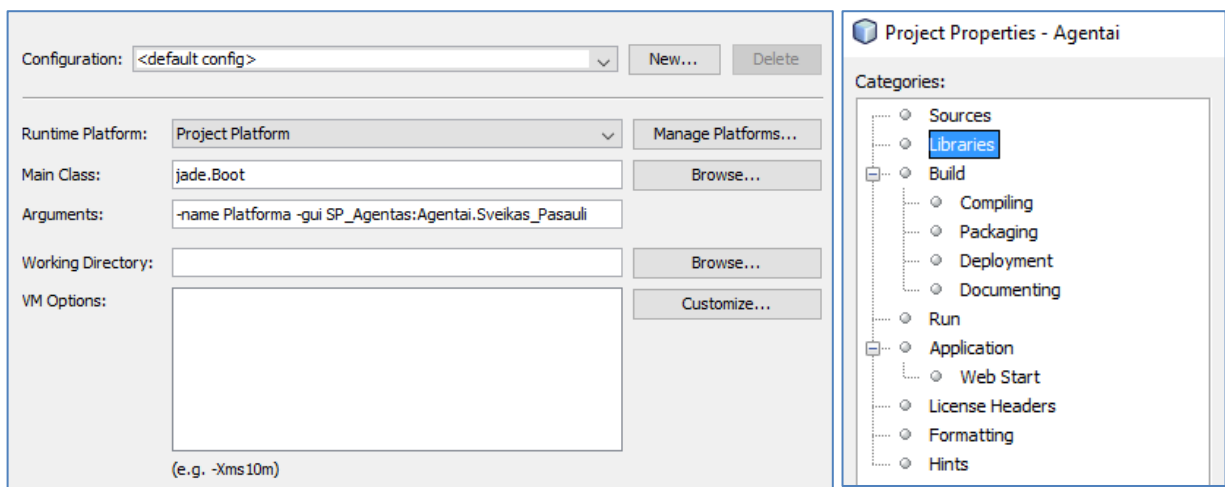
2.2. Nustatykite projekto konfigūraciją: **Run -> Set Project Configuration -> Customize...**



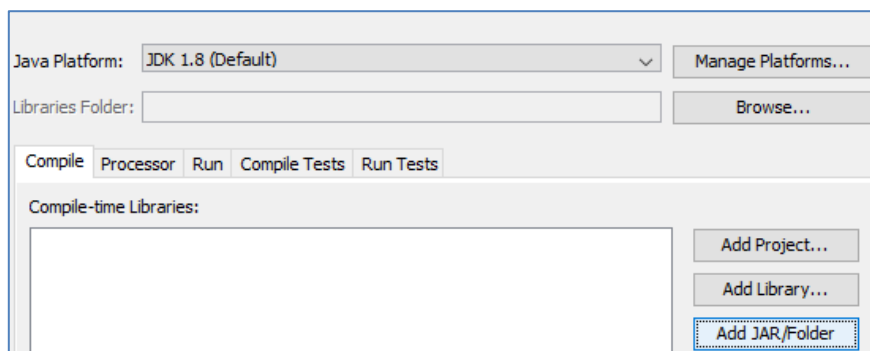
2.3. Laukelyje **Main Class** nurodykite: **jade.Boot**.

Laukelyje **Arguments** nurodykite: **-name Platforma -gui SP_Agentas:Agentai.Sveikas_Pasauli**

2.4. Kairėje lango pusėje pasirinkite kategoriją **Libraries**



2.5. Dešinėje lango pusėje pasirinkite **Add JAR/Folder**



2.6. Pasirinkite **jade.jar** failą, spauskite **open** ir **OK**.

3. „Sveikas_pasauli“ standartinio agento realizavimas

3.1. Atidarykite **Sveikas_pasauli.java**.

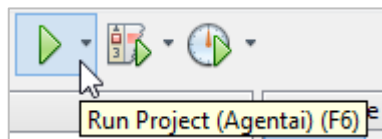
3.2. Importuokite **jade.core.Agent** klasę.

```
import jade.core.Agent;
```

3.3. Sukurkite **Sveikas_pasauli** agentą.

```
public class Sveikas_Pasauli extends Agent
{
    @Override
    protected void setup()
    {
        System.out.println("Sveikas_Pasauli");
        System.out.println("Mano vardas yra - " + getAID().getLocalName());
    }
}
```

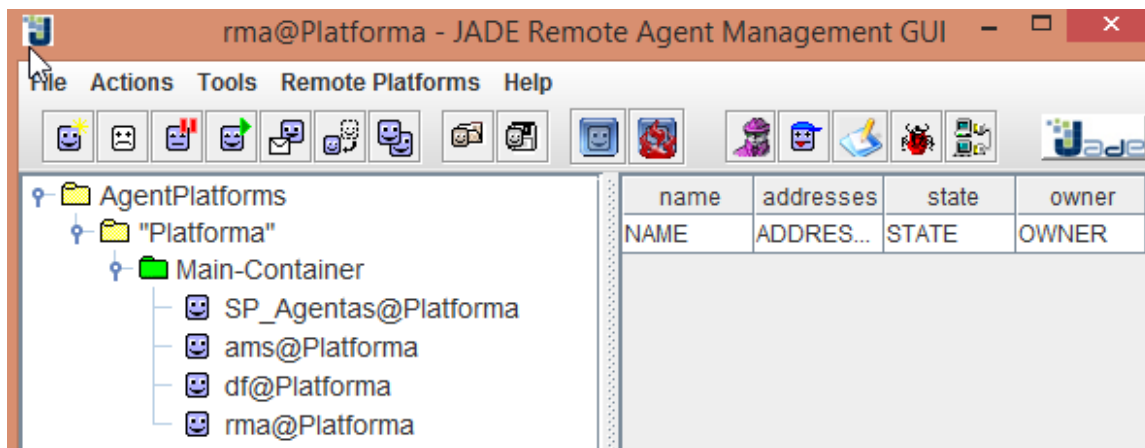
3.4. Paleiskite agentą. Spauskite **run main project** mygtuką.



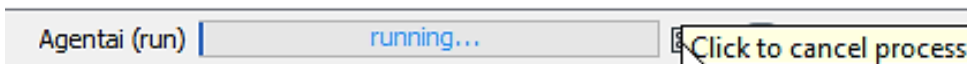
3.5. Išvesties lange matome:

```
Sveikas_Pasauli
Mano vardas yra - SP_Agentas
```

3.6. Automatiškai pasileidžia **Jade Remote Agent Management GUI**.



3.7. Sustabdomė agentą lango apačioje, dešinėje pusėje.



4. Agento elgsenos

4.1. Agento pasisveikinimą realizuokite pridėdami agentui elgseną.

Importuokite elgsenos klasę:

```
import jade.core.behaviours.OneShotBehaviour;
```

```
import jade.core.behaviours.OneShotBehaviour;
```

4.2. `setup()` viduje pridėkite: `addBehaviour(new OneShotBehaviour(this) {`

```
addBehaviour(new OneShotBehaviour(this)
{
    @Override
    public void action()
    {
        System.out.println("Sveikas_Pasauli");
        System.out.println("Mano vardas yra - " + getAID().getLocalName());
    }
});
```

4.3. Paleiskite agentą. Išsiaiškinkite kaip veikia ir ką daro ši elgsena.

4.4. Importuokite elgsenos klasę:

```
import jade.core.behaviours.TickerBehaviour;
```

4.5. `setup()` viduje pridėkite: `addBehaviour(new TickerBehaviour(this, 1000) {`

```
addBehaviour(new TickerBehaviour(this, 1000)
{
    protected void onTick()
    {
        // Programos kodas
    }
});
```

4.6. Papildikite programos kodą, kad agentas atgaliniu skaičiavimu po 10 sekundžių išsijungtų.

4.7. Paleiskite agentą. Rezultatas turėtų gautis panašus į nurodytą apačioje. Išsiaiškinkite kaip veikia ir ką daro ši elgsena.

```
10 sek
9 sek
8 sek
7 sek
6 sek
5 sek
4 sek
3 sek
2 sek
1 sek
0 sek
```

4.8. Importuokite elgsenos klasę: `import jade.core.behaviours.WakerBehaviour;`

4.9. `setup()` viduje pridėkite: `addBehaviour(new WakerBehaviour(this,1000){`

```
addBehaviour(new WakerBehaviour(this,1000)
{
    protected void handleElapsedTimeout()
    {
        // Programos kodas
    }
});
```

4.10. Papildykite programos kodą, kad agentas tik po 15 sekundžių vykdytų **TickerBehaviour** elgseną, naudodami elgseną **WakerBehaviour**.

4.11. Paleiskite agentą. Išsiaiškinkite kaip veikia ir ką daro ši elgsena.

4.12. Importuokite elgsenos klasę:

```
import jade.core.behaviours.CyclicBehaviour;
```

4.13. `Setup()` viduje pridėkite: `addBehaviour(new CyclicBehaviour(this){`

```
addBehaviour( new CyclicBehaviour(this)
{
    public void action()
    {
        // Programos kodas
    }
});
```

4.14. Papildykite programos kodą, kad agentas Išsijungtų papildžius **CyclicBehaviour** naudojant programos fragmentą iš 4,10 punkto.

5. Argumentų eilutė

5.1. Argumentų eilutėje, kuri pavaizduota 2.3 punkte, galima pateikti ir globalius kintamuosius.

```
Arguments: rma -gui SP_Agentas:Agentai.Sveikas_Pasauli(Sveikas pasauli)
```

6. Savarankiško darbo užduotys

Nr.	Sudėtingumas	Užduotis
1		<p>Sukurti agentą, kuris kas 5 sekundes išvestų pranešimą (kurį nurodyti reikia argumentų eilutėje) ir po 30 sekundžių agentas išsijungtų. Laikas turi būti rodomas pranešimų lange.</p> <p>Argumentų eilutė turi atrodyti taip: <code>„<Agento Vardas>(<pranešimas>)“</code></p>
2		<p>Sukurti agentą, kuris argumentų eilutėje nurodytu laiko intervalu išvestų pranešimą: „Sveiki aš agentas vardu: <Agento vardas>“ ir po 30 sekundžių agentas išsijungtų. Laikas turi būti rodomas pranešimų lange.</p> <p>Argumentų eilutė turi atrodyti taip: <code>„<Agento Vardas>(<intervalas>)“</code></p>
3		<p>Sukurti agentą, kuris argumentų eilutėje nurodytu laiko intervalu išvestų pranešimą, kuris taip pnurodytas argumentų eilutėje. Po argumentų eilutėje nurodyto laiko agentas išsijungia. Laikas turi būti rodomas pranešimų lange.</p> <p>Argumentų eilutė turi atrodyti taip: <code>„<Agento Vardas>(<intervalas>,<pranešimas>,<nurodytas laikas>)“</code></p>
4		<p>Sukurti agentą, kuris nurodytą laiko intervalu (kuris nurodytas argumentų eilutėje arba jei jo nenurodėte jis automatiškai turi būti lygus 5) išvestų pranešimą, kuris nurodytas argumentų eilutėje. Po, argumentų eilutėje, nurodyto laiko agentas išsijungia. Laikas turi būti rodomas pranešimų lange.</p> <p>Argumentų eilutė turi atrodyti taip: <code>„<Agento Vardas>(<intervalas>,<pranešimas>,<nurodytas laikas>)“</code> Arba <code>„<Agento Vardas>(<pranešimas>,<nurodytas laikas>)“</code></p>

Labaraotrinio darbo ketvirto skyriaus punktų realizacijos

4.6 Punkto realizacija:

```
addBehaviour(new TickerBehaviour(this, 1000)
{
    protected void onTick()
    {
        if(y == 0)
        {
            System.out.println(y + " sek");
            doDelete();
        }
        else
        {
            System.out.println(y + " sek");
            y = y - 1;
        }
    }
});
```

4.10 Punkto realizacija:

```
addBehaviour(new WakerBehaviour(this, 15000)
{
    protected void handleElapsedTimeout()
    {
        t = true;
    }
});

addBehaviour(new TickerBehaviour(this, 1000)
{
    protected void onTick()
    {
        if(t == true)
        {
            if(y == 0)
            {
                System.out.println(y + " sek");
                doDelete();
            }
            else
            {
                System.out.println(y + " sek");
                y = y - 1;
            }
        }
    }
});
```


4.14 Punkto realizacija:

```
public class Sveikas_Pasauli extends Agent
{
    // kintamieji
    int y = 10;
    int k = 0;
    boolean t = false;
    boolean V = false;

    @Override
    protected void setup()
    {
        addBehaviour(new WakerBehaviour(this, 15000)
        {
            protected void handleElapsedTimeout()
            {
                t = true;
            }
        });

        addBehaviour(new TickerBehaviour(this, 1000)
        {
            protected void onTick()
            {
                if(t == true)
                {
                    if(y == 0)
                    {
                        System.out.println(y + " sek");
                        V = true;
                    }
                    else
                    {
                        System.out.println(y + " sek");
                        y = y - 1;
                    }
                }
            }
        });

        addBehaviour( new CyclicBehaviour(this)
        {
            public void action()
            {
                if(V)
                {
                    doDelete();
                }
            }
        });
    }
}
```