

# Рекомендации по установке и работе с JADE

## Установка JADE с использованием Gradle

### Технические требования

Последующие рекомендации актуальны для следующих инструментов разработки:

- [IntelliJ IDEA](#) – версия 2017 и выше;
- [Java Development Kit](#) – версия 8 и ниже (*Примечание: использование более низких версий на текущий момент обусловлено проблемами в работе системы сборки проектов Gradle с версией JDK 9*).

### Настройка IntelliJ IDEA

На экране запуска **Welcome to IntelliJ IDEA**:

1. Перейти в пункт меню **Configure --> Project Defaults --> Project Structure**. В случае, если на вкладке **Project Settings --> Project** стоит опция **<No SDK>**, выбрать установленную версию JDK.
2. Перейти в пункт меню **Configure --> Plugins**. Установить плагин **Gradle**, если он еще не установлен.

### Создание проекта

1. На экране запуска **Welcome to IntelliJ IDEA** выбрать **Create New Project**.
2. Выбрать тип проекта **Gradle**, в **Project SDK** версию JDK в

соответствии с техническими требованиями, в **Additional Libraries and Frameworks** выбрать пункт Java.

3. Ввести **GroupId** и **ArtifactId**. Например: `ru.spbu` и `jade-test`.
4. В следующем окне выбрать **Gradle JVM** аналогично версии JDK.
5. Настроить путь к проекту (если требуется).

## Подключение JADE

1. Открыть файл **build.gradle**.
2. Подключить репозиторий для скачивания [JADE](#):

```
repositories {
    mavenCentral()
    maven {
        url "http://jade.tilab.com/maven/"
    }
}
```

3. Настроить зависимости:

```
dependencies {
    compile group: 'com.tilab.jade', name: 'jade', version: '4.5.0'
}
```

4. В случае, если при создании проекта не была выбрана опция **Enable Auto-Import**, в правом нижнем углу появится всплывающее окно **Gradle projects need to be imported**.

Необходимо выбрать одну из предлагаемых там опций, например **Import Changes**.

## Создание мультиагентной сети

### Создание точки входа в приложение

Создать класс **.java** для запуска контроллера агентов (класс контроллера будет создан позднее):

```
package ru.spbu.mas;

public class App {
    public static void main(String[] args) {
        MainController mc = new MainController();
        mc.initAgents();
    }
}
```

### Создание простого агента

В **src/main/java/<package-name>** создать новый файл **.java**:

```
package ru.spbu.mas;

import jade.core.Agent;
import java.util.concurrent.TimeUnit;

public class DefaultAgent extends Agent {
```

```

private String[] linkedAgents;
private float number;

@Override
protected void setup() {
    int id = Integer.parseInt(getAID().getLocalName()
);
    System.out.println("Agent #" + id);

    addBehaviour(new FindAverage(this, TimeUnit.SECONDS.toMillis(1)));
}
}

```

В указанном коде были использованы следующие методы JADE:

- getAID() - Method to query the private Agent ID;
- getLocalName() - Method to query the agent local name.

### Добавление поведения агента

В `src/main/java/<package-name>` создать новый файл `.java`:

```

package ru.spbu.mas;

import jade.core.Agent;
import jade.core.behaviours.TickerBehaviour;

public class FindAverage extends TickerBehaviour {

```

```

private final DefaultAgent agent;
private int currentStep;
private final int MAX_STEPS = 10;

FindAverage(DefaultAgent agent, long period) {
    super(agent, period);
    this.setFixedPeriod(true);
    this.agent = agent;
    this.currentStep = 0;
}

@Override
protected void onTick() {
    if (currentStep < MAX_STEPS) {
        System.out.println("Agent " + this.agent.getLocalName() + ": tick=" + getTickCount());
        this.currentStep++;
    } else {
        this.stop();
    }
}
}

```

## Создание контроллера

В `src/main/java/<package-name>` создать новый файл `.java`:

```
package ru.spbu.mas;

import jade.core.Profile;
import jade.core.ProfileImpl;
import jade.core.Runtime;
import jade.wrapper.AgentController;
import jade.wrapper.ContainerController;

class MainController {

    private static final int numberOfAgents = 5;

    void initAgents() {
        // Retrieve the singleton instance of the JADE Runtime
        Runtime rt = Runtime.instance();

        //Create a container to host the Default Agent
        Profile p = new ProfileImpl();
        p.setParameter(Profile.MAIN_HOST, "localhost");
        p.setParameter(Profile.MAIN_PORT, "10098");
        p.setParameter(Profile.GUI, "true");
        ContainerController cc = rt.createMainContainer(p
    );

        try {
            for(int i=1; i <= MainController.numberOfAgen
```

```
ts; i++) {  
    AgentController agent = cc.createNewAgent  
(Integer.toString(i), "ru.spbu.mas.DefaultAgent", null);  
    agent.start();  
}  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}  
}
```

## Запуск проекта

1. В **Edit Configurations** добавить новую конфигурацию **Application**.
2. В **Main Class** записать **ru.spbu.mas.App**, в **Use classpath of module** выбрать **jade-test\_main**. При настройке необходимо учесть собственные названия пакета проекта и т.д.