

# Конфигурационное управление/Практика 2

---

Низамутдинов Степан, ИКБО-20-24

Вариант 16

## Запуск

Запуск приложения с настройками по умолчанию (файл config.yaml):

```
py grapher.py
```

Запуск приложения с заданным файлом настроек:

```
py grapher.py --config <config-file.yaml>
```

```
py grapher.py -c <config-file.yaml>
```

## Этап 5

---

Список изменений:

- Добавлена визуализация графа зависимостей в файл .svg

Настройка визуализации зависимостей пакета Junit

```
version: '0.2'

package-group: org.junit.jupiter
package-name: junit-jupiter-api
package-version: 6.0.0

repository-path: files/repo/
repository-mode: local
output-path: output.svg
ascii-format: none
max-depth: 15
```

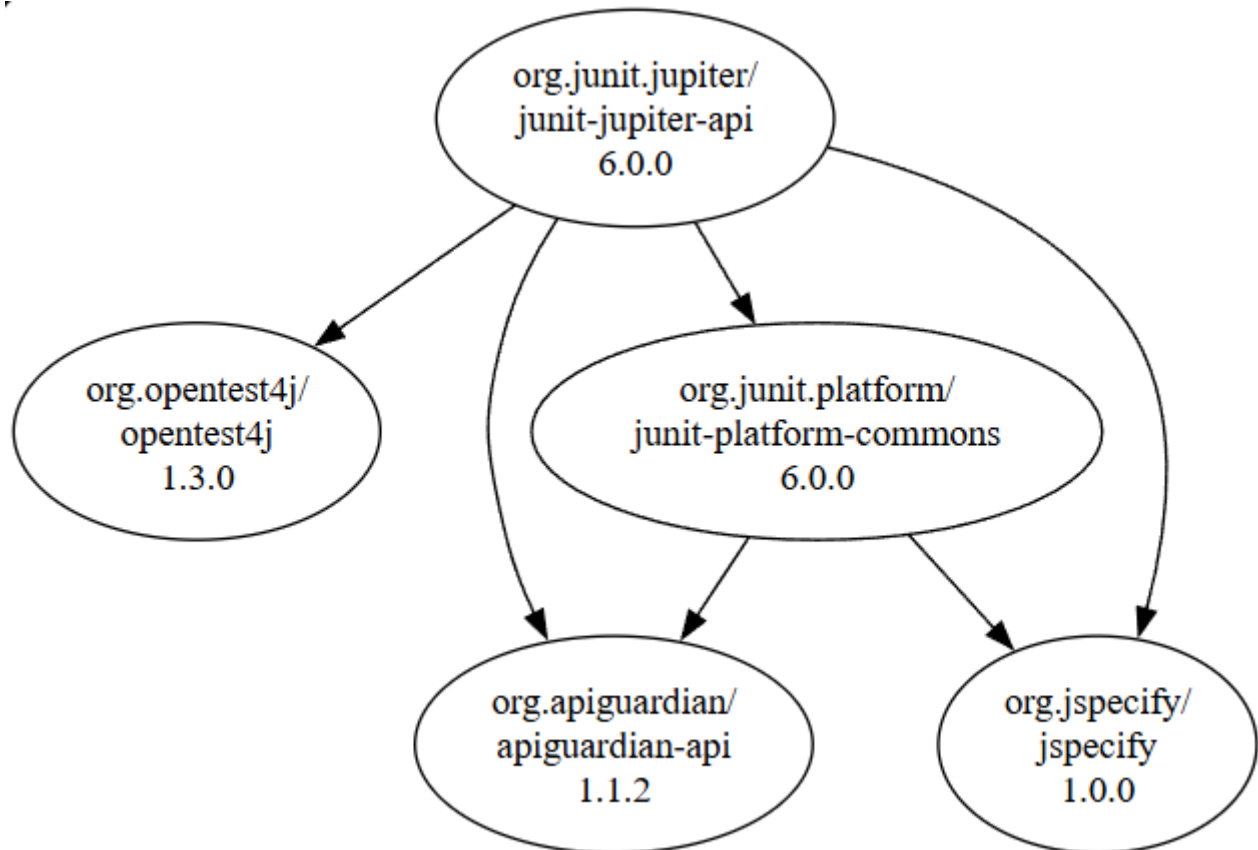
- Добавлен режим вывода графа зависимостей в формате ascii

Варианты настройки `ascii-format`:

- `none` - отсутствие текстового вывода
- `list` - вывод списка зависимостей
- `tree` - вывод `ascii`-дерева зависимостей

## Демонстрация работы:

- Визуализация зависимостей пакета Junit:



- Вывод зависимостей пакета Junit в виде списка:

```
=== Dependency list ===
MavenPackage(org.junit.jupiter/junit-jupiter-api - 6.0.0):
- MavenPackage(org.opentest4j/opentest4j - 1.3.0)
- MavenPackage(org.junit.platform/junit-platform-commons - 6.0.0)
- MavenPackage(org.apiguardian/apiguardian-api - 1.1.2)
- MavenPackage(org.jspecify/jspecify - 1.0.0)

MavenPackage(org.opentest4j/opentest4j - 1.3.0):
-

MavenPackage(org.junit.platform/junit-platform-commons - 6.0.0):
- MavenPackage(org.apiguardian/apiguardian-api - 1.1.2)
- MavenPackage(org.jspecify/jspecify - 1.0.0)

MavenPackage(org.apiguardian/apiguardian-api - 1.1.2):
-
```

```
MavenPackage(org.jspecify/jspecify - 1.0.0):
```

```
-
```

- Вывод зависимостей пакета Scala в формате ascii-дерева:

```
MavenPackage(org.scala-lang/scala3-library_3 - 3.7.3):
├─ MavenPackage(com.github.sbt/junit-interface - 0.13.3):
│   └─ MavenPackage(junit/junit - 4.13.2):
│       ├── MavenPackage(org.hamcrest/hamcrest-core - 1.3)
│       └─ MavenPackage(org.hamcrest/hamcrest-library - 1.3):
│           └─ MavenPackage(org.hamcrest/hamcrest-core - 1.3)
│               └─ ...
└─ MavenPackage(org.scala-sbt/test-interface - 1.0):
    └─ MavenPackage(org.scalatest/scalatest_2.10 - 2.0.M6-SNAP24):
        ├── MavenPackage(org.scala-lang/scala-library - 2.10.0)
        ├── MavenPackage(org.scala-lang/scala-compiler - 2.10.0)
        ├── MavenPackage(org.scalatest/test-interface - 1.0-SNAP3)
        ├── MavenPackage(org.antlr/stringtemplate - 3.2)
        ├── MavenPackage(org.scalacheck/scalacheck_2.10 - 1.10.0)
        ├── MavenPackage(org.easymock/easymockclassexension - 3.1)
        ├── MavenPackage(org.jmock/jmock-legacy - 2.5.1)
        ├── MavenPackage(org.mockito/mockito-all - 1.9.0)
        ├── MavenPackage(org.testng/testng - 6.3.1)
        ├── MavenPackage(com.google.inject/guice - 3.0)
        ├── MavenPackage(junit/junit - 4.10)
        ├── MavenPackage(org.seleniumhq.selenium/selenium-java - 2.31.0)
        ├── MavenPackage(net.sourceforge.cobertura/cobertura - 1.9.1)
        ├── MavenPackage(commons-io/commons-io - 1.3.2)
        ├── MavenPackage(org.eclipse.jetty/jetty-server - 8.1.8.v20121106)
        ├── MavenPackage(org.eclipse.jetty/jetty-webapp - 8.1.8.v20121106)
        ├── MavenPackage(asm/asm - 3.3.1)
        ├── MavenPackage(org.pegdown/pegdown - 1.1.0)
        ├── MavenPackage(org.ops4j.pax.exam/pax-exam-container-native - 2.6.0)
        ├── MavenPackage(org.ops4j.pax.exam/pax-exam-junit4 - 2.6.0)
        ├── MavenPackage(org.ops4j.pax.exam/pax-exam-link-mvn - 2.6.0)
        ├── MavenPackage(org.ops4j.pax.url/pax-url-aether - 1.5.0)
        ├── MavenPackage(org.apache.felix/org.apache.felix.framework - 4.0.3)
        ├── MavenPackage(org.eclipse.tycho/org.eclipse.osgi - 3.7.0.v20110613)
        ├── MavenPackage(org.slf4j/slf4j-api - 1.7.1)
        ├── MavenPackage(ch.qos.logback/logback-core - 1.0.7)
        └─ MavenPackage(ch.qos.logback/logback-classic - 1.0.7)
└─ MavenPackage(org.scala-lang/scala-library - 2.13.16)
```

## Этап 4

Список изменений:

- Добавлен поиск обратных зависимостей пакета

Пример поиска обратных зависимостей пакета:

```
Building dependency graph...
```

```
Enter package to analyse: org.apiguardian/apiguardian-api - 1.1.2
```

```
Inverse dependencies for MavenPackage(org.apiguardian/apiguardian-api - 1.1.2):
```

- MavenPackage(org.junit.platform/junit-platform-suite-engine - 1.10.2)
- MavenPackage(org.junit.platform/junit-platform-engine - 1.10.2)
- MavenPackage(org.junit.platform/junit-platform-suite-api - 1.10.2)
- MavenPackage(org.junit.platform/junit-platform-suite-commons - 1.10.2)
- MavenPackage(org.junit.platform/junit-platform-commons - 1.10.2)
- MavenPackage(org.junit.platform/junit-platform-launcher - 1.10.2)

## Демонстрация работы:

- Нахождение обратных зависимостей пакета log4j (версии 1.2.15):

```
Inverse dependencies for MavenPackage(log4j/log4j - 1.2.15):
```

- MavenPackage(org.springframework/spring - 2.5.6)
- MavenPackage(org.hibernate/hibernate-commons-annotations - 3.0.0.ga)

- Нахождение обратных зависимостей пакета guice (версии 2.0):

```
Inverse dependencies for MavenPackage(com.google.inject/guice - 2.0):
```

- MavenPackage(org.testng/testng - 6.1.1)
- MavenPackage(org.testng/testng - 6.8.13)
- MavenPackage(org.testng/testng - 5.14.2)
- MavenPackage(org.testng/testng - 6.8)
- MavenPackage(org.testng/testng - 5.14.1)
- MavenPackage(org.testng/testng - 6.8.1)
- MavenPackage(org.testng/testng - 6.8.5)
- MavenPackage(org.testng/testng - 5.12.1)
- MavenPackage(org.testng/testng - 6.8.8)
- MavenPackage(org.testng/testng - 5.14.10)
- MavenPackage(org.testng/testng - 6.5.2)

## Этап 3

---

### Список изменений:

- Добавлен режим работы с локальным репозиторием

Пример настроек режима работы:

```
repository-path: tests/repo/  
repository-mode: local
```

- Реализовано построение графа зависимостей

Для указания глубины анализа используется настройка `max-depth` (0 - поиск только прямых зависимостей). При построении графа используется алгоритм обхода в ширину.

## Демонстрация работы:

- Построение графа зависимостей пакета Junit (режим удаленного репозитория):

```
$ py grapher.py -c tests/junit-test.yaml
```

```
Fetches 1 package (depth 0)  
Fetches 4 packages (depth 1)  
  
MavenPackage(org.junit.jupiter/junit-jupiter-api - 6.0.0):  
- MavenPackage(org.opentest4j/opentest4j - 1.3.0)  
- MavenPackage(org.junit.platform/junit-platform-commons - 6.0.0)  
- MavenPackage(org.apiguardian/apiguardian-api - 1.1.2)  
- MavenPackage(org.jspecify/jspecify - 1.0.0)  
  
MavenPackage(org.opentest4j/opentest4j - 1.3.0):  
-  
  
MavenPackage(org.junit.platform/junit-platform-commons - 6.0.0):  
- MavenPackage(org.apiguardian/apiguardian-api - 1.1.2)  
- MavenPackage(org.jspecify/jspecify - 1.0.0)  
  
MavenPackage(org.apiguardian/apiguardian-api - 1.1.2):  
-  
  
MavenPackage(org.jspecify/jspecify - 1.0.0):  
-
```

- Построение графа зависимостей пакета Junit (режим локального репозитория):

```
$ py grapher.py -c tests/junit-test-local.yaml
```

```

  Fetched 1 package (depth 0)
  Fetched 4 packages (depth 1)

  MavenPackage(org.junit.jupiter/junit-jupiter-api - 6.0.0):
  - MavenPackage(org.opentest4j/opentest4j - 1.3.0)
  - MavenPackage(org.junit.platform/junit-platform-commons - 6.0.0)
  - MavenPackage(org.apiguardian/apiguardian-api - 1.1.2)
  - MavenPackage(org.jspecify/jspecify - 1.0.0)

  MavenPackage(org.opentest4j/opentest4j - 1.3.0):
  -

  MavenPackage(org.junit.platform/junit-platform-commons - 6.0.0):
  - MavenPackage(org.apiguardian/apiguardian-api - 1.1.2)
  - MavenPackage(org.jspecify/jspecify - 1.0.0)

  MavenPackage(org.apiguardian/apiguardian-api - 1.1.2):
  -

  MavenPackage(org.jspecify/jspecify - 1.0.0):
  -
```

## Этап 2

---

### Список изменений:

- Версия файла настроек обновлена до 0.2

Новые настройки:

- `package-group` - группа пакета
- `package-version` - версия пакета

Пример нового файла настроек:

```
%YAML 1.2
---
version: '0.2'

package-group: org.junit.jupiter
package-name: junit-jupiter-api
package-version: 6.0.0

repository-path: https://repo.maven.apache.org/maven2/
repository-mode: url
output-path: output.png
ascii-format: default
max-depth: 15
```

- Добавлена поддержка формата пакетов Maven

Пример получения прямых зависимостей из [Центрального Репозитория Maven](#):

```
$ py grapher.py -c config.yaml
```

```
MavenPackage(org.junit.jupiter/junit-jupiter-api - 6.0.0):  
- MavenPackage(org.opentest4j/opentest4j - 1.3.0)  
- MavenPackage(org.junit.platform/junit-platform-commons - 6.0.0)  
- MavenPackage(org.apiguardian/apiguardian-api - 1.1.2)  
- MavenPackage(org.jspecify/jspecify - 1.0.0)
```

## Демонстрация работы:

Получение зависимостей пакета Spring:

```
$ py grapher.py -c tests/files/spring-test.yaml
```

```
MavenPackage(org.springframework.boot/spring-boot-starter-web - 3.5.4):  
- MavenPackage(org.springframework.boot/spring-boot-starter - 3.5.4)  
- MavenPackage(org.springframework.boot/spring-boot-starter-json - 3.5.4)  
- MavenPackage(org.springframework.boot/spring-boot-starter-tomcat - 3.5.4)  
- MavenPackage(org.springframework/spring-web - 6.2.9)  
- MavenPackage(org.springframework/spring-webmvc - 6.2.9)
```

Получение зависимостей пакета Junit:

```
$ py grapher.py -c tests/files/junit-test.yaml
```

```
MavenPackage(org.junit.jupiter/junit-jupiter-api - 5.13.4):  
- MavenPackage(org.opentest4j/opentest4j - 1.3.0)  
- MavenPackage(org.junit.platform/junit-platform-commons - 1.13.4)  
- MavenPackage(org.apiguardian/apiguardian-api - 1.1.2)
```

Получение зависимостей пакета Scala:

```
$ py grapher.py -c tests/files/scala-test.yaml
```

```
MavenPackage(org.scala-lang/scala3-library_3 - 3.7.3):  
- MavenPackage(com.github.sbt/junit-interface - 0.13.3)  
- MavenPackage(org.scala-lang/scala-library - 2.13.16)
```

## Этап 1

---

### Список изменений:

- Добавлено считывание настроек из файла

Доступные настройки:

- `version` - версия файла настроек
- `package-name` - название пакета
- `repository-path` - путь к репозиторию
- `repository-mode` - режим работы с репозиторием (`local` | `url`)
- `output-path` - путь к файлу вывода
- `ascii-format` - формат вывода в ascii-дереве (`default`)
- `max-depth` - максимальная глубина анализа зависимостей

Пример файла настроек:

```
%YAML 1.2  
---  
version: '0.1'  
package-name: package  
repository-path: repo.txt  
repository-mode: local  
output-path: output.png  
ascii-format: default  
max-depth: 15
```

### Демонстрация работы:

Пример обычного запуска:

```
$ py grapher.py
```

```
Version: 0.1  
Package name: package  
Repository path: repo.txt  
Repository mode: local  
Output path: output.png
```



```
Ascii format: default  
Max depth: 15
```

### Примеры обработки ошибок:

- Ошибка в синтаксисе файла

```
$ py grapher.py -c wrong-config.yaml
```

```
Error! Config file is not a valid yaml file
```

- Ошибка в параметре максимальной глубины

```
$ py grapher.py -c depth-test.yaml
```

```
Error! Invalid max depth: -1
```

- Несоответствующая версия файла

```
$ py grapher.py -c version-test.yaml
```

```
Error! Config version mismatch (999.0 != 0.1)
```