

**Гуляев Г.М.**  
**Современные технологии  
программирования**

**Лекция 7. Система сборки sbt и IntelliJ Idea**

Центр компетенции СПО Алтайского края



## Simple build tool (sbt)

- ❖ Сборка большого Scala проекта внутри IDE (NetBeans, IntelliJ IDEA, Eclipse) занимает значительное время, поэтому для сборки используется sbt ([www.scala-sbt.org](http://www.scala-sbt.org)).
- ❖ Особенности sbt:
  - простая конфигурация (мало или нет для простых проектов)
  - основа языка конфигов - **Scala**
  - смешанные проекты (**Java/Scala**)
  - поддержка внешних Java библиотек (**.jar**)
  - генерация документации **ScalaDoc**
  - **Ivy** для управления зависимостями
  - Запуск **REPL** в контексте проекта
- ❖ Установка:
  - скачиваем архив **sbt.tgz** с ресурса <http://www.scala-sbt.org/release/docs/Getting-Started/Setup.html>
  - распаковываем в любой каталог
  - добавляем путь на подкаталог **bin** в переменную окружения **PATH**
- ❖ Результат:
  - команда **sbt** доступна из любого каталога файловой системы

# Среда разработки IntelliJ IDEA

- ❖ Для ввода текстов программ и быстрой навигации по файлам проекта используется IDE **IntelliJ IDEA**.
- ❖ **Установка:**
  - скачиваем последнюю версию IntelliJ IDEA community edition:  
<http://www.jetbrains.com/idea/download/index.html>
  - распаковываем в любой каталог
  - размещаем на рабочем столе запускающий скрипт, типа:  

```
#!/bin/sh  
cd /home/user/soft/idea-12/bin  
sh idea.sh
```
- ❖ Для поддержки **scala** и **sbt** установим плагины:
  - Главное меню → **File** → **Settings** → **Plugins** → **Browse repositories** → ввод в строке поиска **scala** → выбор **scala** и **sbt**
  - скачивание выбранных плагинов и их установка
- ❖ **Порядок работы:**
  - каталог проекта **sbt** и его структура (подкаталоги и конфиги) создаются вручную (копируется из предыдущих проектов)
  - запуск **sbt**, генерация файлов для **IDEA**, проект открывается в **IDEA**

# Структура подкаталогов sbt-проекта

**<build.sbt>**

**project/**

**<build.properties, plugins.sbt>**

**src/**

**main/**

**resources/**

**<иконы и другие ресурсы>**

**scala/**

**<исходный код на scala>**

**java/**

**<исходный код на java>**

**test/**

**resources**

**<ресурсы для тестов>**

**scala/**

**<исходный код на scala для тестов>**

**java/**

**<исходный код на java для тестов>**

❖ **Не используемые каталоги могут отсутствовать.**

# Конфигурационные файлы проекта

## ❖ **build.sbt:**

```
import AssemblyKeys._
```

```
name := "prog"
```

```
version := "1.0"
```

```
scalaVersion := "2.10.2"
```

```
resolvers += "typesafe" at "http://repo.typesafe.com/typesafe/snapshots"
```

```
libraryDependencies ++= Seq(  
  "org.scala-lang" % "scala-swing" % "2.10.2",  
  "mysql" % "mysql-connector-java" % "5.1.18"  
)
```

```
assemblySettings
```

# Конфигурационные файлы проекта

## ❖ **build.properties:**

```
sbt.version=0.12.4
```

## ❖ **plugins.sbt:**

```
resolvers += Resolver.url("artifactory",  
url("http://scalasbt.artifactoryonline.com/scalasbt/sbt-plugin-  
releases")) (Resolver.ivyStylePatterns)
```

```
addSbtPlugin("com.github.mpeltonen" % "sbt-idea" % "1.5.1")
```

```
addSbtPlugin("com.eed3si9n" % "sbt-assembly" % "0.9.1")
```

## Команды sbt и плагинов

- ❖ > **clean** - очистка (удаление подкаталога target)
- ❖ > **update** - удовлетворение зависимостей
- ❖ > **reload** - перезагрузка файлов конфигурации
- ❖ > **compile** - компиляция проекта
- ❖ > **run** - запуск на выполнение (могут быть аргументы)
- ❖ > **package** - упаковка в jar
- ❖ > **packageDoc** - упаковка документации в jar
- ❖ > **packageSrc** - упаковка исходников в jar
- ❖ > **test** - прогон тестов
- ❖ > **console** - вызов консоли REPL в контексте проекта
- ❖ > **exit, quit, Ctrl-D** - выход
- ❖ > **assembly** - сборка всех зависимостей в один jar для jvm (плагин)
- ❖ > **gen-idea** - генерация файлов для IDEA (плагин)

# Задания для самостоятельной работы

- ❖ Установить на своем компьютере **sbt** и **IntelliJ IDEA**
- ❖ Используя **sbt**, **IntelliJ IDEA** и пакет **scala.swing**, написать приложение с оконным интерфейсом для нахождения повторяющихся частей текста. Текст может быть введен пользователем или загружен с диска. Подзадачи:
  - а) вводится любая фраза - ищется число ее повторов в тексте
  - б) вычисляется статистика повторяющихся слов, результат выводится в порядке убывания числа повторений



Спасибо за внимание!

[www.altailand.ru](http://www.altailand.ru)

