

Scala - spotkanie 1

1. Instalacja

- a. Sprawdzić czy zainstalowana jest Java w wersji JDK - powinny być dostępne narzędzia java oraz javac. Jeśli brak javac - patrz przypis na końcu dokumentu.
- b. Wejść na stronę <http://www.scala-lang.org/> oraz zapoznać się z nią. Szczególną uwagę należy zwrócić na [Tour of Scala](#) oraz dokumentację [API](#).c
- c. Na zakładce **Downloads** wybrać **DOWNLOAD SBT**
- d. Odnośnik prowadzi do strony narzędzia **SBT (Scala Build Tool)** będącego podstawą dla zarządzania i budowania kodu w Scali.
- e. Pobrać lokalnie plik znajdujący się części All Platforms - (z uwagi na to że nie mamy praw administracyjnych w systemie możemy uruchomić narzędzie lokalnie na koncie bieżącego użytkownika)
- f. Rozpakować zawartość pliku lokalnie najlepiej do Katalogu Apps. Jeśli taki nie istnieje trzeba go utworzyć.
- g. Dla sbt trzeba nadać prawa do wykonania: `chmod +x sbt`
- h. Dodać ścieżkę programu do zasobów systemowych - przypisy 1 lub 2 w zależności od systemu operacyjnego.

2. Utworzenie pierwszego programu

- a. Utworzyć katalog dla projektu np: Lab1
c
- b. Utworzyć plik build.sbt o zawartości:
- c. Wewnątrz src/main/scala utworzyć plik Pierwszy.scala o zawartości:

```
object Pierwszy extends App {println("Witaj świecie!!")}
```
- d. Uruchomić program poprzez: `sbt run`

3. Praca w REPL (ang. read-eval-print loop)

- a. W konsoli przejść od katalogu utworzonego projektu
- b. Uruchomić z konsoli polecenie: `sbt console`
- c. Dalsze przykłady działania scali można wykonywać w REPL obserwując działanie testowanych poleceń.

4. Praca z IDE - na stronie scali w zakładce downloads znajduje się odnośnik do preferowanego w trakcie pracy ze scala zintegrowanego środowiska programistycznego (ang. Integrated Development Environment) jakim jest IntelliJ (ok. 540 MB). Oczywiście nie jest to jedyny wybór. Na stronie <http://scala-ide.org> można znaleźć IDE oparte na Eclipse (ok 280 MB) - także plugin do eclipse. Dostępne są także pluginy dla innych popularnych edytorów jak: atom, vs, SublimeText.

Zadania do wykonania

1. Napisz funkcję rysującą prostokąt o zadanych wymiarach, który składa się z samych gwiazdek. Na przykład po wywołaniu tej funkcji z parametrami 8 i 3 powinniśmy otrzymać:

```
* * * * *
* * * * *
* * * * *
```

2. Napisz program obliczający wartość wyrażenia:

$$w = \sqrt[3]{2 \log 521 + \frac{750000000}{128 \cdot 422}}$$

Wskazówki:

- Liczbę 750000000 można zapisać w postaci: `val x = 7.5e8;`
 - Importować bibliotekę `scala.math._`
3. Napisz program, który znajduje wszystkie trzycyfrowe liczby, których suma sześcianów poszczególnych cyfr jest równa danej liczbie. (Wskazówki: Skorzystaj z operatorów arytmetycznych `%` i `/` dla argumentów całkowitych, gdzie:
 - `%` - operator modulo - reszta z dzielenia. Na przykład wartością wyrażenia `937%10` jest liczba 7;
 - `/` - operator dzielenia całkowitego. Na przykład wartością wyrażenia `937/10` jest liczba 93 (ale wartością `937/10.0` jest liczba 93.7).
 - proszę spróbować nie korzystać z instrukcji pętli tylko z innych własności scali

Przypis 1.

Konfiguracja zmiennych środowiskowych w systemie Windows

1. Przejsć do Start -> Control Panel -> System
2. Przejsć do zakładki Advanced Settings i kliknąć przycisk Environmental Variables
3. Na górnej liście sprawdzić czy zdefiniowana jest zmienna JAVA_HOME jeśli jej nie ma to trzeba ją utworzyć.
4. Jako wartość zmiennej JAVA_HOME podać ścieżkę do katalogu instalacji java jdk. W salach 209/210 D1 lokalizacja ta to: c:\Program Files\Java\jdk1.8.0_xx\
5. Jeśli nie istnieje utworzyć drugą zmienną środowiskową PATH
6. Wartość zmiennej PATH to %PATH%;%JAVA_HOME%\bin
7. Zamknąć okna przyciskiem OK
8. Sprawdzić w systemie czy po tej operacji dostępne jest narzędzie javac poprzez otwarcie konsoli (polecenie `cmd` lub `powershell`) oraz podaniu polecenia `javac -version` - powinna wyświetlić się wersja kompilatora.

Przypis 2.

Dodanie katalogu do ścieżek dostępu w systemie Linux:

1. Jeśli nie istnieje utworzyć katalog bin (w konsoli: `mkdir ~/bin`)
2. Edytować (utworzyć jeśli go brak) plik profilowy: `mcedit ~/.profile` (kropka przed nazwą wskazuje na plik ukryty)
3. Na końcu pliku dopisać (choć prawdopodobnie jest już ustawione):

```
if [ -d "$HOME/bin" ] ; then PATH="$HOME/bin:$PATH" fi
```
4. Zapisać

Przypis 2a.

Utworzenie linku symbolicznego do sbt

1. `ln -s ~/App/sbt/bin/sbt ~/bin/sbt`