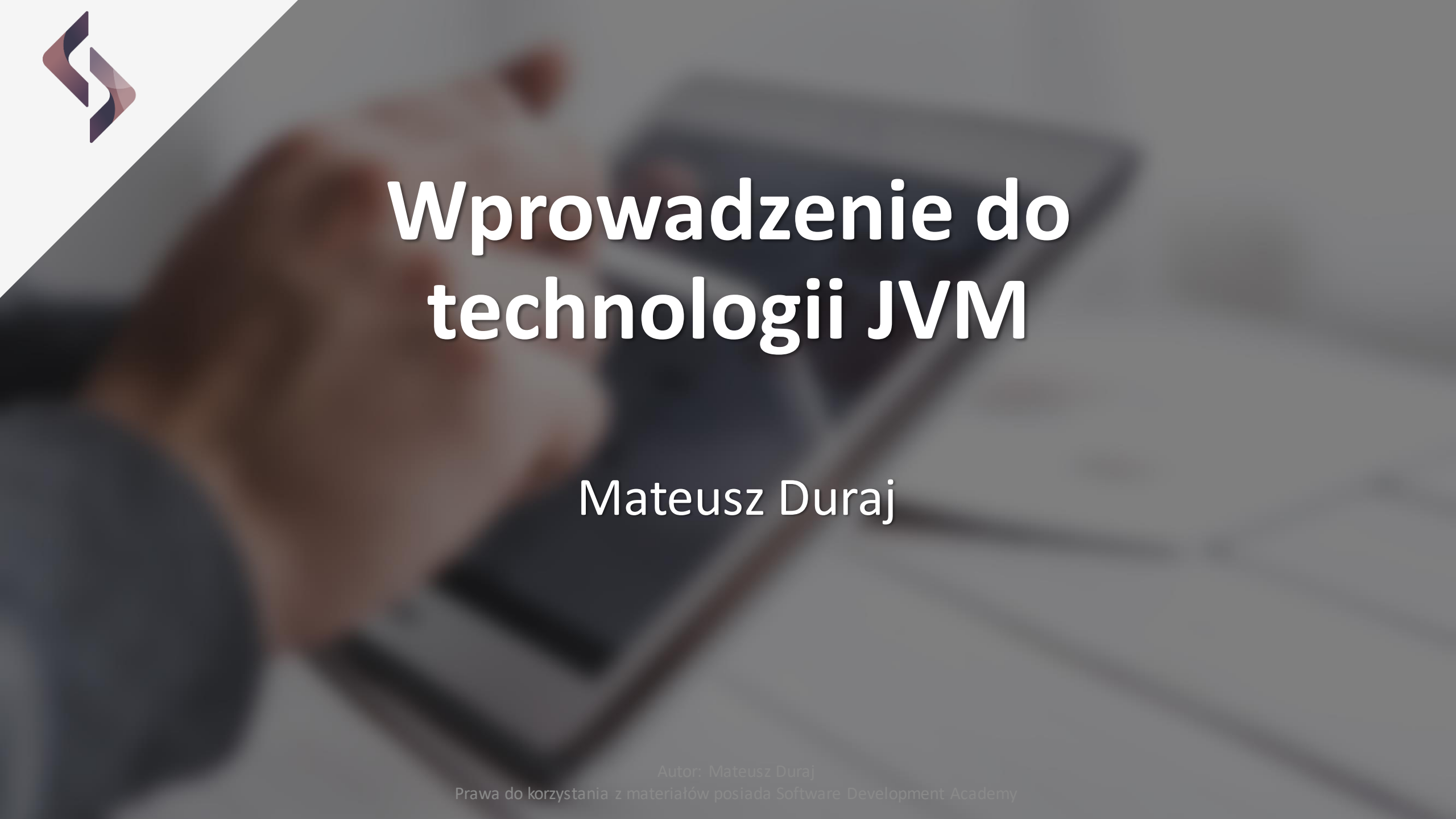


Wprowadzenie do technologii JVM

Mateusz Duraj

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Agenda

1. JVM
2. Byte code
3. ClassLoader
4. JIT
5. Thread introduction
6. Java memory model
7. GC & Memory Leaks
8. JDK Tools

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

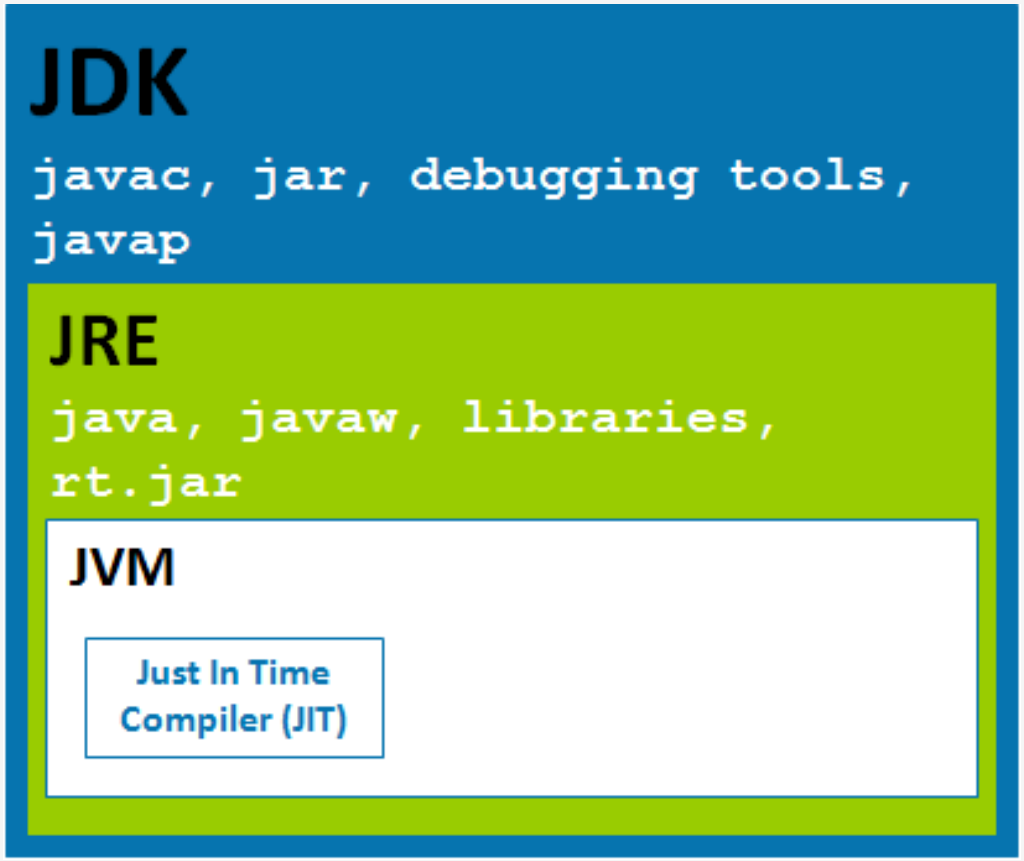
JVM – co to takiego ?

- JVM to akronim dla Java Virtual Machine, tj. Wirtualna Maszyna Javy.
- Aby aplikacja napisana w języku Java mogła być uruchomiona, pliki zawierające kod źródłowy muszą być skompilowane.
- Kompilacja polega na przetłumaczeniu programu napisanego w Javie na kod wykonywalny, tzw. Bytecode.
- Kompilację wykonujemy przy użyciu specjalnego narzędzia, tzw. kompilatora.

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

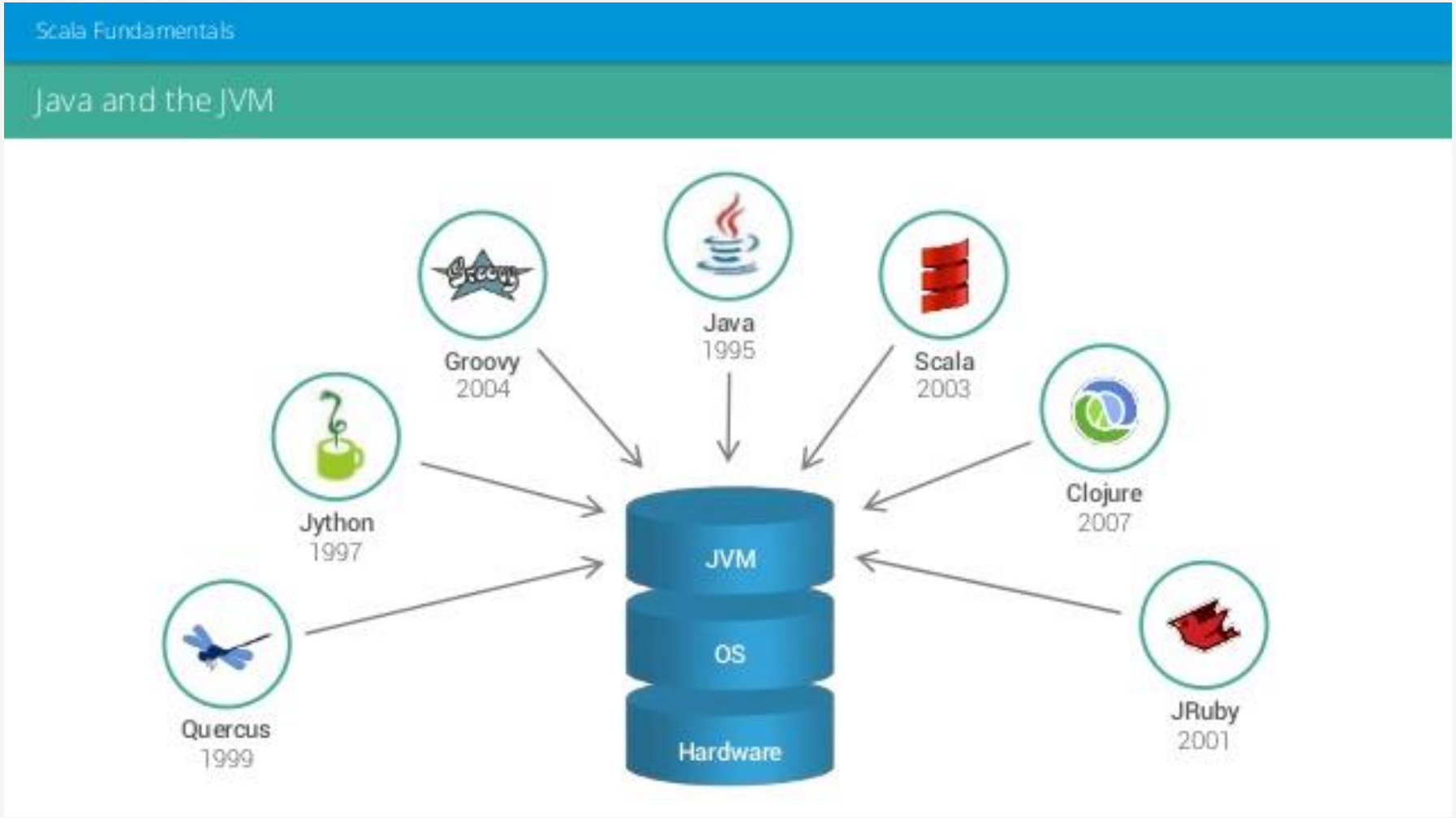
JVM – co to takiego ?

- JDK
- JRE
- JVM



Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JVM – co to takiego ?



Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JVM – lista przykładowych maszyn wirtualnych

HotSpot – oficialna dystrybucja, od 11 płatna dla użytku komercyjnego

OpenJDK – równoległe rozwijana przez Oracla

DoppioJVM - napisana w TypeScript, można odpalać w przeglądarce

leJOS – maszyn wirtualna dla Lego Mindstorms ☺

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Hotspot vs OpenJDK

- Maszyna HotSpot bazuje w dużej mierze na OpenJDK
- Zawiera dodatkowo np. Java Flight Recorder (profiler dla javy) w wersji komercyjnej

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Life cycle

source -> javac ->bytecode

bytecode ->classloader ->interpreter

interpeter -> JIT-> optimized native code

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Zadanie 1

Pobierz projekt jvm2 z repo

Zweryfikuj nazwę implementacji JVM zainstalowanej na Twoim systemie.
W tym celu wykorzystaj properties:

- java.vm.name
- java.home
- java.version
- java.vendor
- java.specification.vendor

Wartość w/w kluczy wypisz na konsolę(otwórz klasę **JvmConfig**)

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Byte code

- Kod maszynowy

  https://en.wikipedia.org/wiki/Java_bytecode_instruction_listings
- Instrukcje
  - ładujące i zapisujące (np. aload_0, istore)
  - arytmetyczne i logiczne (np. ladd, fcmpl)
  - zmieniające typ (np. i2b, d2i)
  - tworzące obiekt i manipulujące nim (np. new, putfield)
  - przenoszące sterowanie (np. ifeq, goto)
  - wywołujące metodę i wracające z niej (np. invokespecial, areturn)

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Zadanie 2 Generowanie kod bajtowego

- W klasie JavapTrain:
- Z poziomu CMD(Windows) wejdź do katalogu projektu(package pl.sdacademy.jvm)
- Wykonaj polecenie javac JavapTrain.java
- Wykonaj polecenie javap –c JavaTrain.class > JavaTrain.bc
- Dla poleceń javac i javap sprawdz dostępne opcje z użycie parametru -help

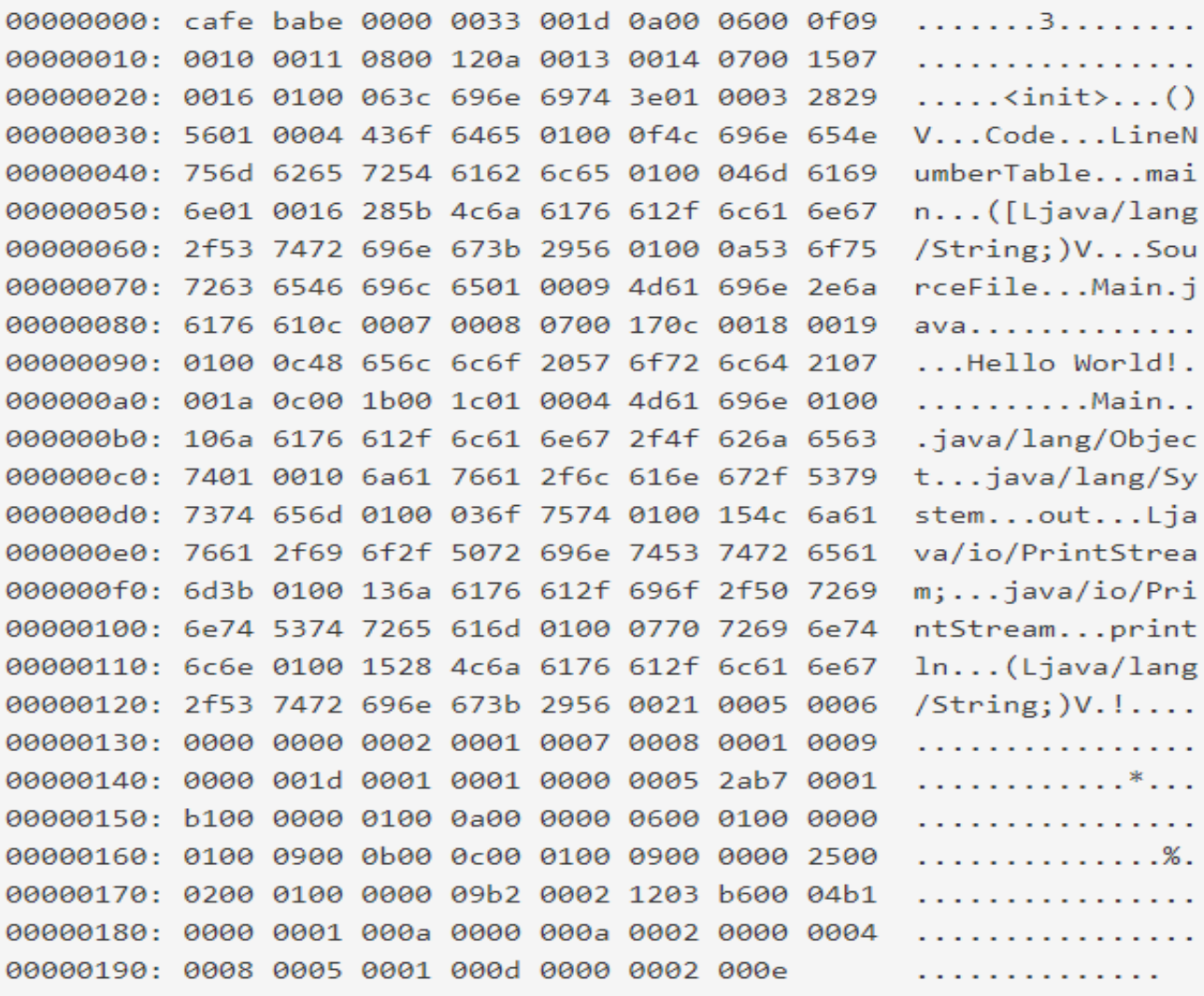
Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

CLASS

- Zawiera meta informacje np. versje javy, superclass, interfaces…
- Informacje przechowywane są w formacie HEX
- Szczegółowy opis formatu pliku.class:
  https://codecouple.pl/2018/06/15/co-kryje-plik-class/

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

.class wersja HEX



Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

CLASSLOADER

Ładuje dynamicznie klasy –

JVM mówi że „potrzebuje takiej klasy”

Czyli mamy pierwszy etapy source->javac->bytecode

Kolejny : classloader -> … ?

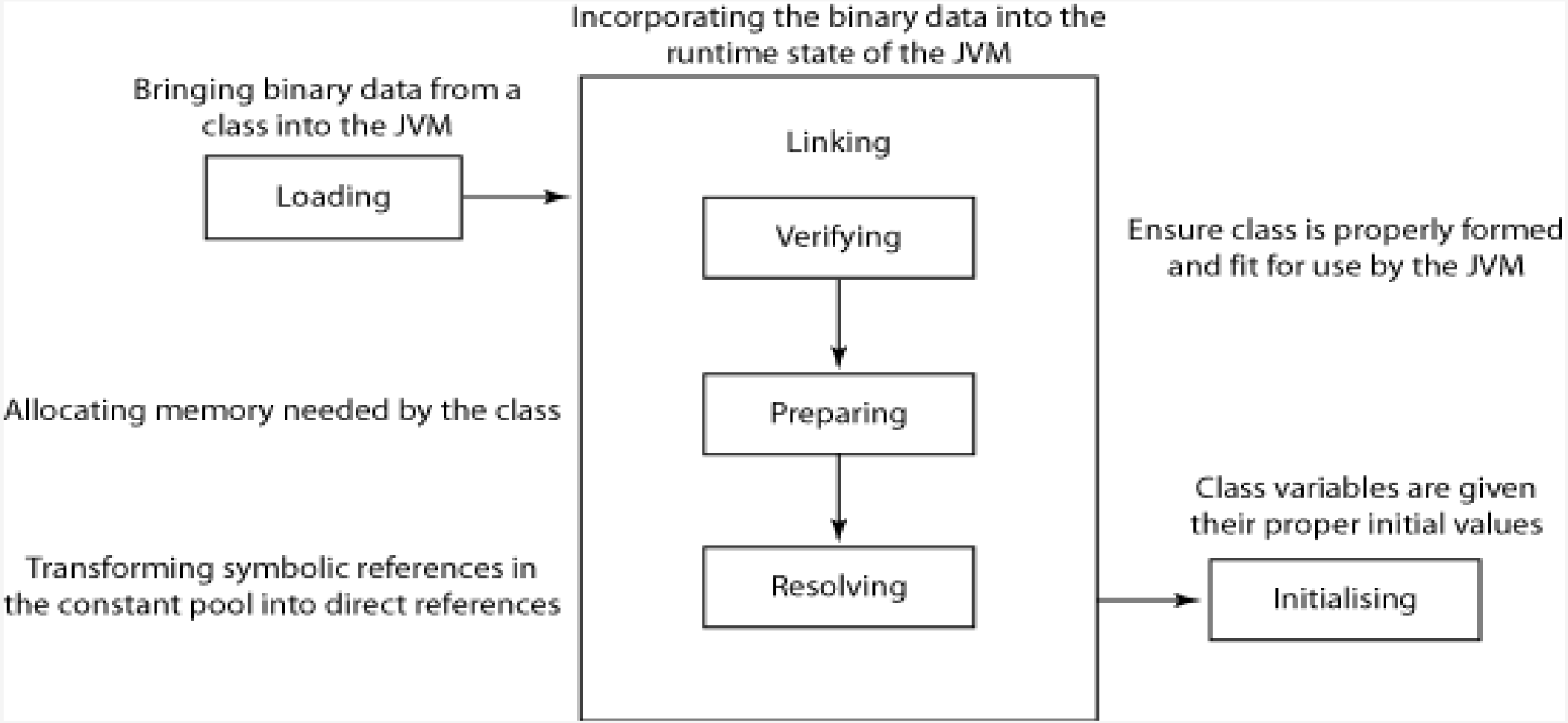
Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

CLASSLOADER - FAZY

- Ładowanie – odczyt plików .class
- Linkowanie
  - Weryfikacja poprawności kodu bajtowego (czy np. nie rozszerzamy klasę final)
  - Alokowanie pamięci (czyli np. w klasie mamy tablice)
  - Wiązanie obiektów (np. mamy dziedziczenie)
- Inicjalizacja statycznych pół klasy

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

CLASSLOADER - FAZY



Prawa do korzystania z materiałów posiada Software Development Academy

CLASSLOADER – faza Loading

- Ładowanie kodu bajtowego do pamięci
- Lokalizacja innych klasy (na podstawie classpaths)
- Bootstrap class loader dla ładowania klas z rt.jar
- Extension class loader dla ładowania klas z $JAVA_HOME/lib/ext
- System class loader dla ładowania klas dla naszej aplikacji (classpath)
- Na tym etapie może pojawić się ClassNotFoundException, NoClassDefFoundError

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

CLASSLOADER – faza Linking

- Weryfikacja poprawności kodu bajtowego (czy np. nie rozszerzamy klasę final)
- Alokowanie pamięci (czyli np. w klasie mamy tablice)
- Wiązanie obiektów (np. mamy dziedziczenie)
- Sprawdzanie poziomów dostępu

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

CLASSLOADER – faza Initialization

- Wykonanie metod inicjalizacyjnych na klasach i interfejsach
- Inicjowanie pól ich wartościami domyślnymi
- Inicjowanie super klas, jeżeli do tej pory nie zostały zainicjowane
- Obiekt klasy jest gotowy do użycia w programie

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Zadanie 3 ClassLoader

- Z poziomu cmd otwórz katalog z klasą **`ClassLoader`**
- `Wykonaj polecenie java –verbose ClassLoader`

Prawa do korzystania z materiałów posiada Software Development Academy

INTERPETER

- Java jest językiem interpretowanym , a więc kompilowany do bytecode, a potem jest to interpretowane przez JVM
- Hot spot – interpreter javy cały czas analizuje kod i np. weryfikuje jak często wywoływana jest dana metoda

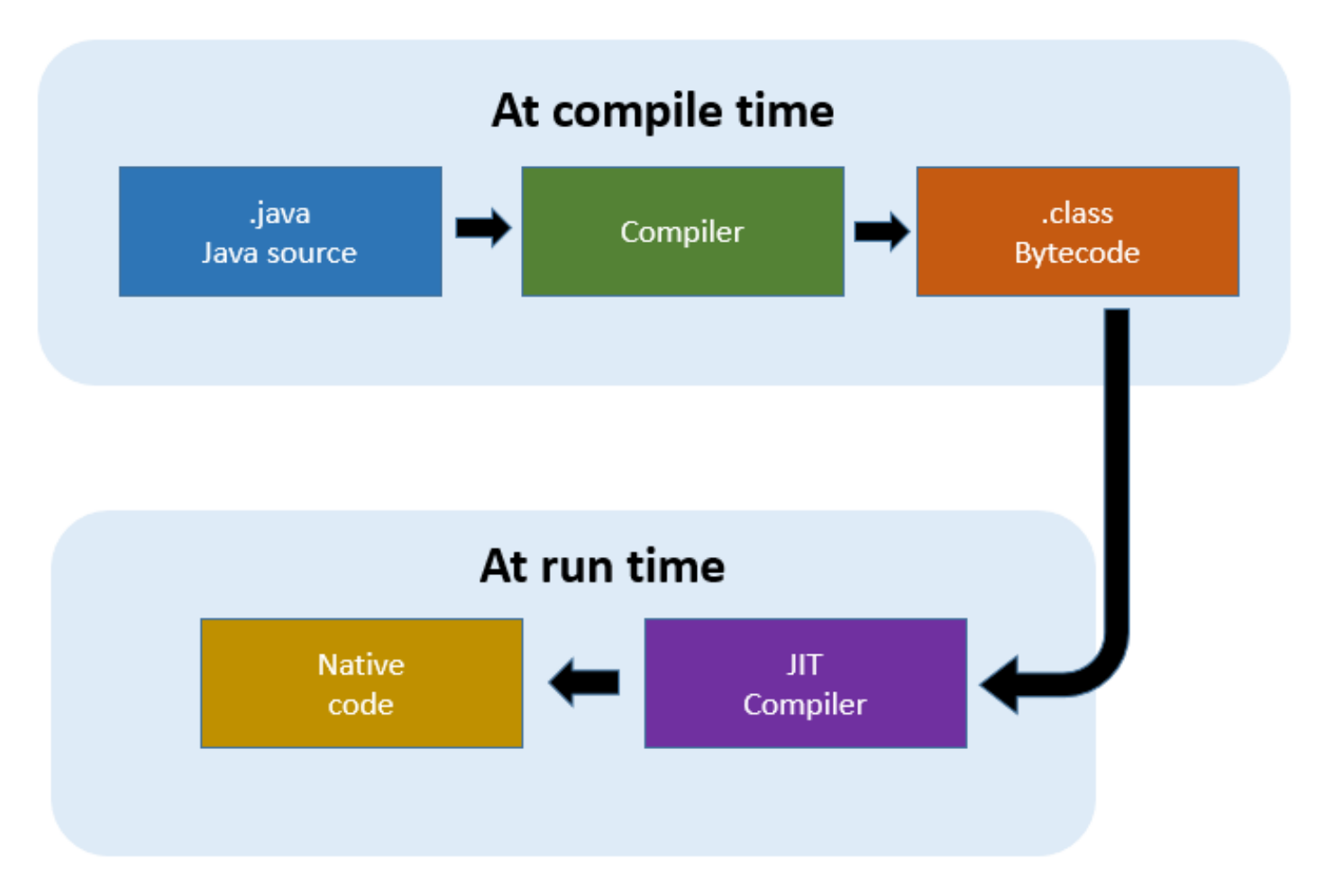
Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JIT (Just-In-Time)

- Kod po fazie „interpretowania”, jest analizowany poprzez wbudowany mechanizm optymalizacji jit
- Średnio jit potrafi zoptymalizować kod do 20 razy szybciej(przy konwersji do kodu natywnego)
- JIT działa w wersjach client i serwer (determinuje to konfiguracja sprzetowa)

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JIT (Just-In-Time)



Zródło: https://aboullaite.me/understanding-jit-compiler-just-in-time-compiler/

Prawa do korzystania z materiałów posiada Software Development Academy

JIT (Just-In-Time) Wybrane optymalizacje

- Inlining (zagnieżdżanie metod)
- Loop unrolling
- Dead-code elimination (eliminacja martwego kodu)

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JIT (Just-In-Time) Inlining

```
public String getValueFromSupplier(Supplier < String > supplier) {
 return supplier.get();
}

public String businessMethod(String param) {
 Supplier < String > stringSupplier = () -> ("my" + param);
 return getValueFromSupplier(stringSupplier);
}

// zostanie zamieniona na

public String businessMethod(String param) {
 Supplier < String > stringSupplier = () -> ("my" + param);
 return stringSupplier.get();
}
```


Prawa do korzystania z materiałów posiada Software Development Academy

JIT (Just-In-Time) Loop unrolling

```
for (int i = 0; i < N; i++) {
 S(i);
}

// Zostanie zamienione na

for (int i = 0; i < N; i += 4) {
 S(i);
 S(i + 1);
 S(i + 2);
 S(i + 3);
}
```

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JIT (Just-In-Time) Dead code elimination

```
public String businessMethod(String param) {
 Supplier < String > supplier = new StringSupplier("my" + param);
 if (supplier == null) {
  throw new IllegalArgumentException("Supplier cannot be null");
 }
 return supplier.get();
}

// Zostanie zamienione na

public String businessMethod(String param) {
 Supplier < String > supplier = new StringSupplier("my" + param);
 return supplier.get();
}
```

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JIT – parametryzacja JVM

- -XX:+UnlockDiagnosticVMOptions
- -XX:+LogCompilation - hotspot_pid.log – drukuje informacje o poziomach client/serwer
- -XX:+TraceClassLoading

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Question

Java jest językiem kompilowany czy interpretowanym ?

Prawa do korzystania z materiałów posiada Software Development Academy

Kompilacja poziomowa (TIERED COMPILATION)

Domyślnie od JDK8, wcześniejsze wersje -XX:+TieredCompilation

0 : kod intepretowany
1: prosty kod natywny C1(client)
2: ograniczony kod natywny C1
3: pełny kod natywny C1
4: kod natywny C2(serwerowy)

Prawa do korzystania z materiałów posiada Software Development Academy

String , JVM a IMMUTABILITY

- Dlaczego klasa String jest final ?
- Róznica pomiędzy new String(”Message ”) a ”Message”

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Zadanie 5 IMMUTABILITY

- W projekcie jvm utwórz package immutability
- Potrzebne będą 3 klasy : Book, BookCollection, Price oraz Main
- Klasa Book zawiera pola id, title, author, price (typu Price)
- Klasa Price zawiera pole value, oraz currency
- Klasa BookCollection zawiera 2 metody:
  - findBookByTitle
  - printAllBooks

  Zaprojektuj strukturę klas oraz implementacje w/w metod

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Question - Przekazywanie zmiennych

- Java przekazuje zmienne przez wartość czy referencje?
- Słowo kluczowe final w odniesieniu do zmiennej

Prawa do korzystania z materiałów posiada Software Development Academy

Zadanie 6

- W projekcie jvm odszukaj klasy `JavaStack`
- Uruchom prosty program(prawy przycisk i ,Run `JavaStack`)
- Zlokalizuj PID Procesu JVM za pomocą polecenia jps –l
  - (lista parametrów https://docs.oracle.com/javase/7/docs/technotes/tools/share/jps.html)

- Uruchom narzędzie jvisualvm.exe, znajdujące się katalogu JDK/bin
- Na podstawie PID uzyskanego wcześniej, sprawdź zawartość sterty

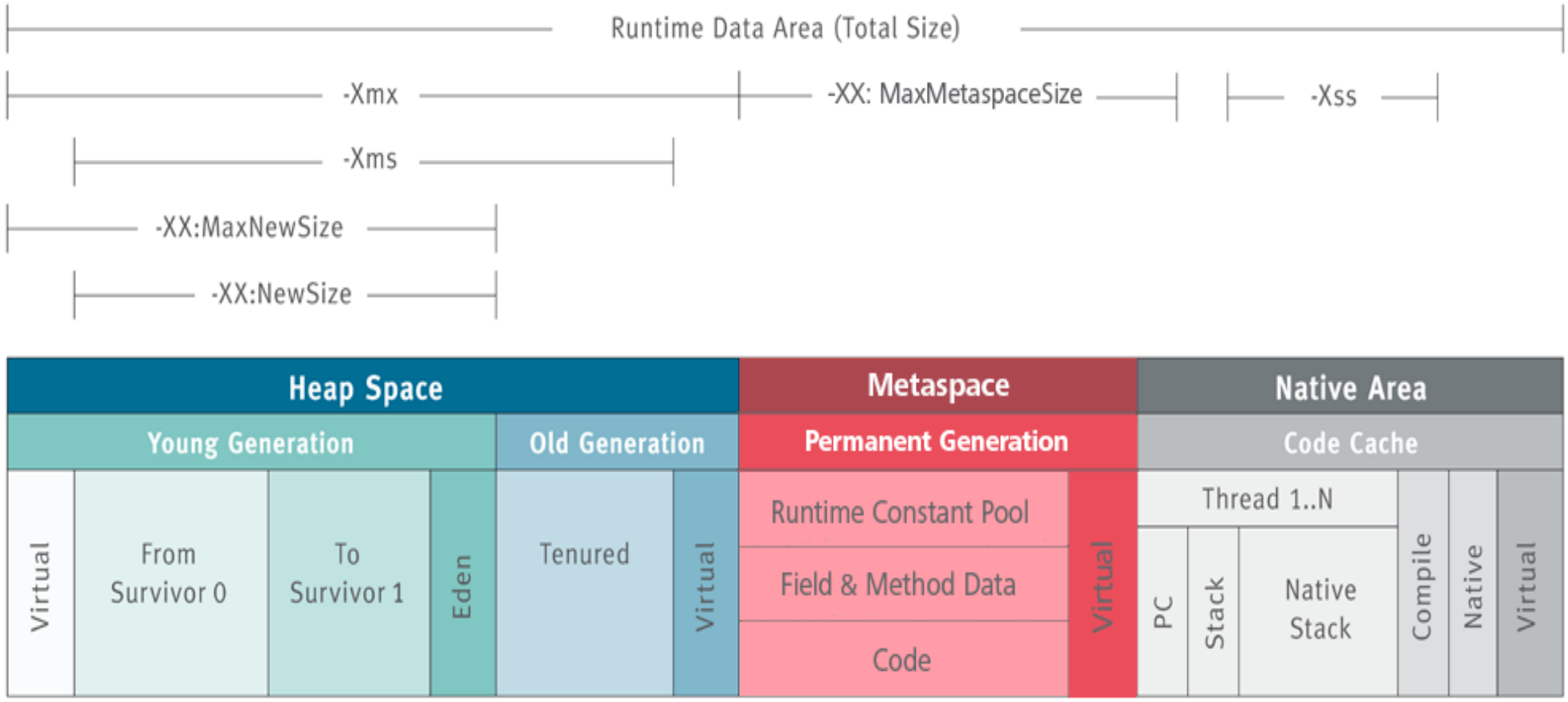
Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Threads & Synchronization ***

- W Javie mamy 2 rodzaje wątków :
  - stworzone przez nasz (new Thread())
  - utworzone przez jvm (np.GC)
- Java memory model opisuje jak działa współdzielenie zasobów przez wątki
- java.util.concurrent

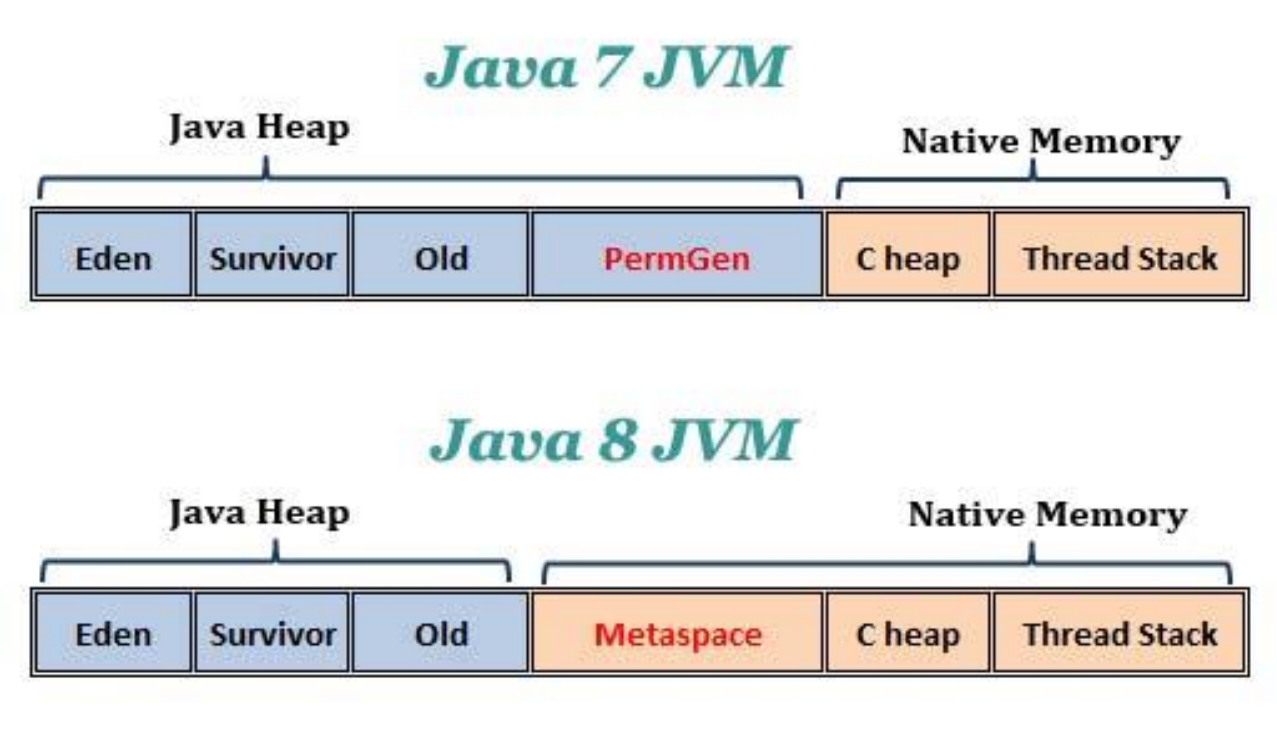
Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JAVA MEMORY MODEL



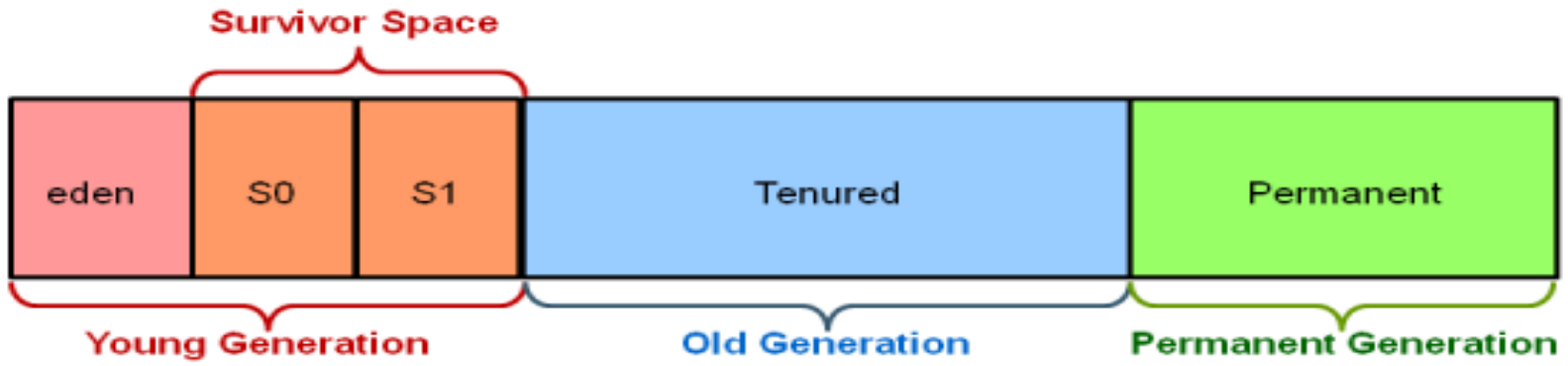
Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JAVA MEMORY MODEL



Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Garbage collector



źródło:
http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html

Prawa do korzystania z materiałów posiada Software Development Academy

Garbage collector - tuning

- -Xms – max heap size
- -Xms starting heap size
- -XX:MaxPermSize set size of the permgen
- -verbose:gc – print to the console when a garbage collection take place
- -Xmn set the size of the young generation

- -XX:HeapDumpOnOutOfMemory – creates a heap dump file

Prawa do korzystania z materiałów posiada Software Development Academy

Garbage collector - algorytmy

- Serial – gdy wątek GC startuje, aplikacja jest zatrzymywana, dobry dla mały aplikacji, wybór –XX: +UseSerialGC
- Parallel – osobny wątek gc – gdy mamy kilka CPU, dobry dla dużych aplikacji –XX:UseParallelGC
- The CMS Collector – dla aplikacji >100MB, z wieloma procesorami, większość wykonania gc / praca aplikacji współbieżnie
- The G1 Collector – zalecany dla serwerów, do obsługi powyżej >4GB

- java -XX:+PrintCommandLineFlags –version < wypisuje defaultowy gc

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Memory blocks

PC REGISTER(program counter – czyli miejsce aktualnie wykonywanej operacji)

FRAME(np. wywołanie nowej metody)

STACK(odkładamy ramki) - składający się z ramek

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

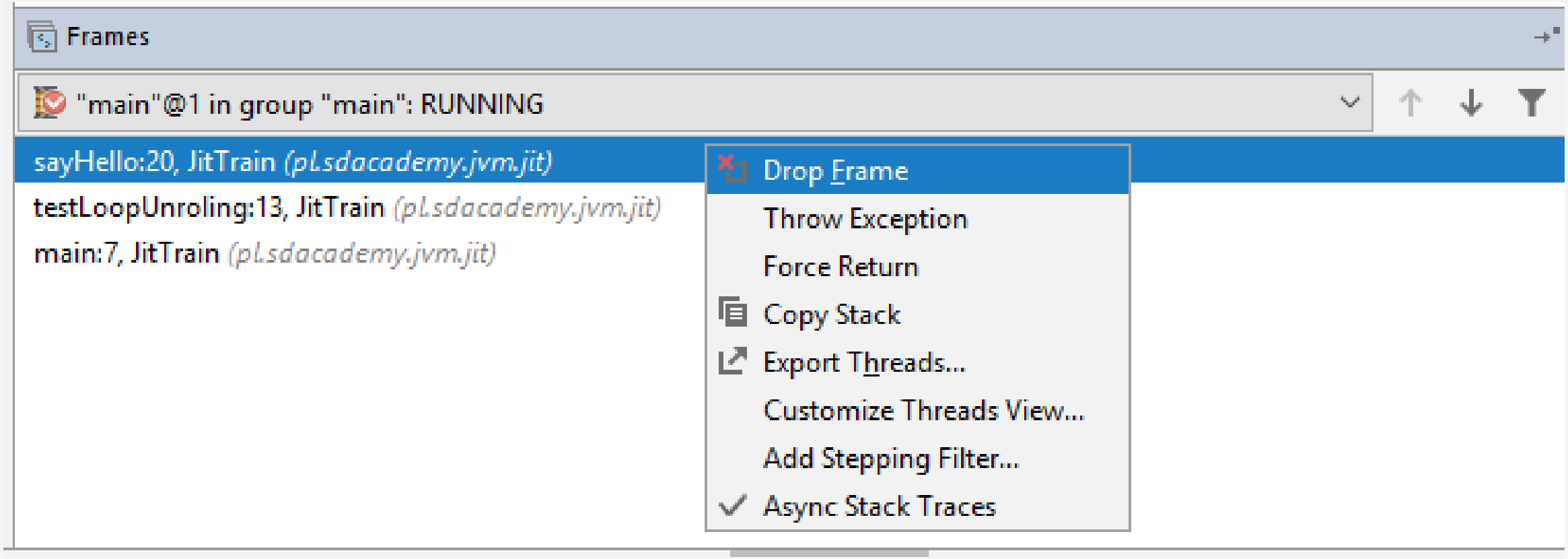
Stack trace

```
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Exception in thread "main" java.lang.NullPointerException: Cannot be null
    at pl.sdacademy.jvm.jit.JitTrain.sayHello(JitTrain.java:20)
    at pl.sdacademy.jvm.jit.JitTrain.testLoopUnroling(JitTrain.java:13)
    at pl.sdacademy.jvm.jit.JitTrain.main(JitTrain.java:7)

Process finished with exit code 1
```

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Debugging



Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Question

Stack vs Heap – różnice ?

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JAVA RLZ!

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Memory Leak (pl.sda.jvm.memory.gc)

W Javie wycieki powinny nie występować dzięki JVM i zarządzaniu pamięcią przez GC – jednak należy uważać ☺

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

JMH Java Microbenchmark Harness

- JMH służy do tworzenie benchmarków w Javie czyli badania wydajności fragmentów naszej aplikacji.

- http://hg.openjdk.java.net/code-tools/jmh/file/tip/jmh-samples/src/main/java/org/openjdk/jmh/samples/ - przykłady użycia

Prawa do korzystania z materiałów posiada Software Development Academy

Narzędzia – pozwalające na pracę z JVM

- **Javap** – generuje czytelny byte code
- **Jstack** – zrzuty wątków
- **Visual vm** – podpinamy się do dzialajce aplikacji, i sprawdaamy tzn telemetrie
- **Mission control**- podpinamy się do dzialajce aplikacji, i sprawdzamy tzn. metryki (np.zajetosc pamięci)
- **Jprofiler** – płatny , kombajn do analizy aplikacji , bardzo dokladna np. gdzie są tworzone obiekty, graf obiektów, ile duplkowany stringów, największe obiekty. Polecam wykorzystać triala jeśli chccie sprawdzić jak działa java
- **Java object layout** – drukuje jak wygląda layout klasy, np. ile zajmują pamięci klasy, róznice pomiędzy boolean a Boolean.

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Chcę więcej , co dalej ?

JDK do pobrania jako projekt:

http://hg.openjdk.java.net/jdk8/jdk8

http://tutorials.jenkov.com/java-performance/jmh.html

https://programistanaswoim.pl/wp-content/uploads/2017/07/jvm1.pdf

https://bottega.com.pl/pdf/materialy/jvm/jvm2.pdf

Autor: Mateusz Duraj
Prawa do korzystania z materiałów posiada Software Development Academy

Zródła

http://java.meritcampus.com

https://www.slideshare.net/AlfonsoRuzafaMolina/scala-fundamentals

https://www.stechies.com/difference-between-permgen-metaspace/

https://www.ibm.com/developerworks/library/j-dclp1/index.html

https://blog.takipi.com/java-on-steroids-5-super-useful-jit-optimization-techniques/

Prawa do korzystania z materiałów posiada Software Development Academy

Zródła

http://docs.oracle.com/javase/specs/jvms/se8/html/. - specyfikacja JVM

Prawa do korzystania z materiałów posiada Software Development Academy