

The background is a solid blue gradient. Overlaid on this are numerous thin, white, curved lines that flow from the left side towards the right, creating a sense of motion and depth. These lines are more densely packed in some areas, forming a wave-like pattern that peaks towards the right side of the image.

JAVA SE 8 PROGRAMMER II

BARTOSZ ANDREATTO



Programista w Banku Pekao S.A z bogatym doświadczeniem w zakresie technologii back-end'owych wykorzystujących wirtualną maszynę Javy. Pasjonat rozwiązań opartych na ekosystemie Spring oraz rozwiązań Oracle. Certyfikowany programista Java. Uwielbia nauczać oraz dzielić się wiedzą.

LinkedIn: <https://www.linkedin.com/in/bartosz-andreatto-02b03413a/>

e-mail: bandreatto@gmail.com



ZAKRES SZKOLENIA

- Budowa klas w Javie
- Zaawansowane aspekty budowy klas w Javie
- Kolekcje i typy generyczne
- Interfejsy funkcyjne wyrażeń lambda
- Java Stream API
- Wyjątki i asercje
- API dotyczące daty / czasu w Javie SE 8
- Podstawy Java IO
- Plikowe I/O w Javie (NIO.2)
- Wielowątkowość
- JDBC
- Lokalizacja



ZAKRES SZKOLENIA

- Java SE 8 Programmer I Exam Number: 1Z0-808

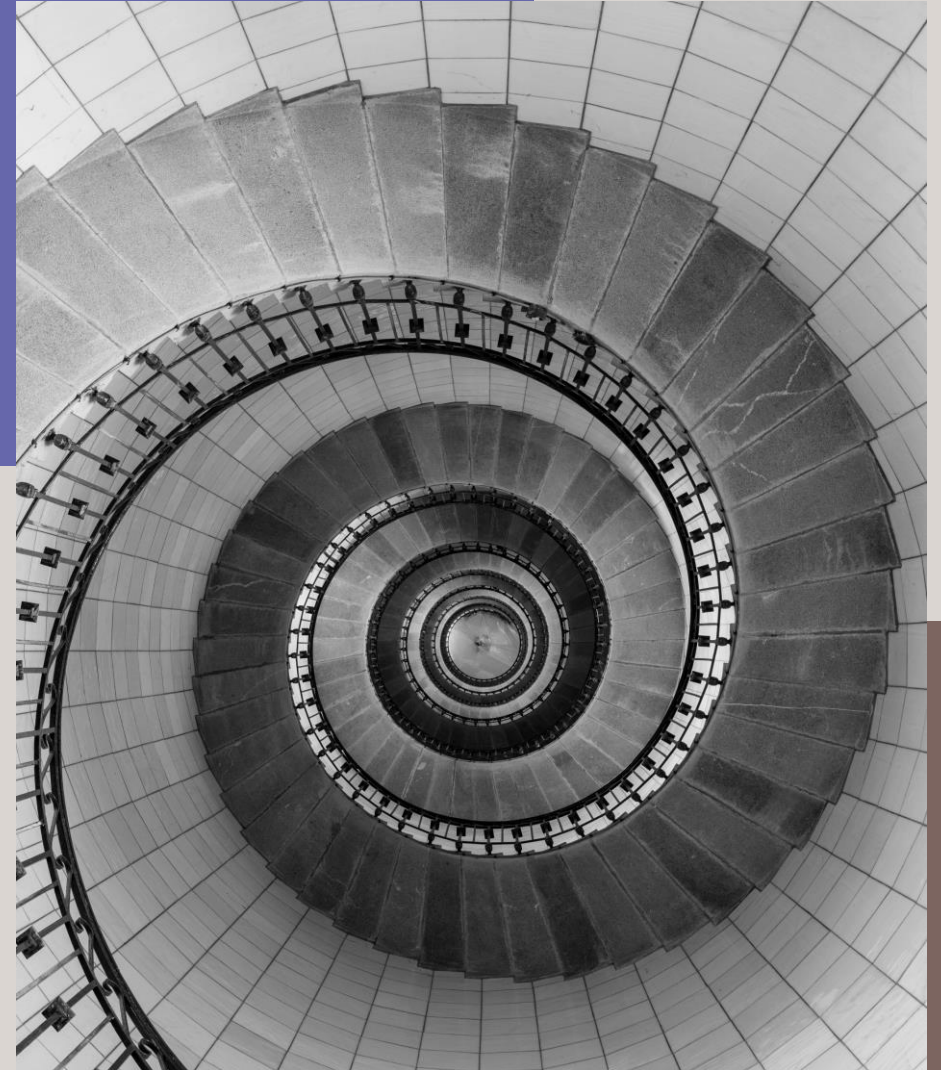
https://education.oracle.com/java-se-8-programmer-i/pexam_1Z0-808

- liczba pytań: **56**
- czas trwania: 120 minut
- pytania wielokrotnego wyboru
- próg: 65%

- Java SE 8 Programmer II Exam Number: 1Z0-809

https://education.oracle.com/java-se-8-programmer-i/pexam_1Z0-808

- liczba pytań: **68**
- czas trwania: 120 minut
- pytania wielokrotnego wyboru
- próg: 65%



ORGANIZACYJNIE

- Repozytorium GIT:

https://github.com/bandreatto/1z0_808_9/tree/slupsk_2022_12

- 28 (7 x 4) godzin lekcyjnych
- Przerwy: ~co 90 minut



PRZYGOTOWANIE

- <https://enthuware.com/java-certification-mock-exams/oracle-certified-professional/java-se-8-1z0-809>
- <https://www.whizlabs.com/ocjp-scjp/>
- <https://www.coderanch.com/wiki/659977/Ocjp-Faq>
- OCA / OCP Java SE 8 Programmer Certification Kit
<https://www.amazon.pl/OCA-OCP-Java-Programmer-Certification/dp/1119272092>
- OCA: Oracle Certified Associate Java SE 8 Programmer I Study Guide:
<https://www.amazon.pl/OCA-Certified-Associate-Programmer-1Z0-808/dp/1118957407>
- OCP: Oracle Certified Professional Java SE 8 Programmer II Study Guide
<https://www.amazon.pl/OCP-Certified-Professional-Programmer-1Z0-809/dp/1119067901/>
- OCA / OCP Java SE 8 Programmer Practice Tests: Exam 1Z0-808 and Exam 1Z0-809
<https://www.amazon.pl/OCA-Java-Programmer-Practice-Tests/dp/111936339X/>



ZAKRES SZKOLENIA – DZIEŃ 1

- Budowa klas w Javie

enkapsulacja; dziedziczenie; modyfikatory dostępu; polimorfizm; metody: toString, equals, hashCode; wzorec singleton; obiekty immutable; byty statyczne; bloki inicjalizacyjne

- Zaawansowane aspekty budowy klas w Javie

klasy i metody abstrakcyjne; słowo kluczowe final; klasy wewnętrzne; enumeracje;

interfejsy: deklaracja, implementacja, dziedziczenie; adnotacja @Override; tworzenie i użycie wyrażeń lambda

- Kolekcje i typy generyczne

klasy generyczne; typy: ArrayList, TreeSet, TreeMap, ArrayDeque; interfejsy: Comparator, Comparable; przetwarzanie kolekcji za pomocą strumieni i filtrów; metoda forEach; interfejs Stream; filtrowanie kolekcji z użyciem wyrażeń lambda; referencje do metod



KLASA – POJĘCIE PODSTAWOWE

Klasy w programowaniu obiektowym służą do opisywania otaczających nas przedmiotów, zdarzeń, czynności, stanów oraz relacji między opisywanymi przedmiotami. Takie typy zdefiniowane za pomocą klas nazywamy **typami złożonymi** lub też **typami referencyjnymi**.

Klasy posiadają dwie podstawowe składowe:

- **pola** – zmienne lub stałe opisujące obiekt danej klasy
- **metody** – operacje, które udostępnia nasza klasa

KLASA – PRZYKŁAD 1

Przypuśćmy, że chcemy opisać film jako klasę. Z jakich zatem cech (pól) może się składać klasa opisująca nasz film? Nic prostszego: tytuł, rok produkcji, opis, aktorzy, itp. Te wszystkie wymienione cechy to nic innego jak pola wchodzące w skład klasy Film. Zostały jeszcze czynności (metody), np.: "odtwórz film" itp.

```
1 public class Movie { // (1)
2     private String title; // (2)
3     private String description; // (3)
4     private int productionYear; // (4)
5
6     public void play() { // (5)
7         // ciało metody
8     }
9 }
```

DZIĘKUJĘ ZA UWAGĘ

