

20 WRZEŚNIA

19:00-20:00

DARMOWY WEBINAR

**5 prostych kroków, żeby zdobyć
pracę jako Java Developer w kilka
miesięcy nawet jeśli masz mało
wolnego czasu**

ZACZYNAMY O 19:00



BARTEK KALKA

JAVAREADY.PL

JAK SŁYCHAĆ?



PIERWSZY RAZ?



KTO ZACZYNA?



**JESTEŚ TUTAJ
DLATEGO, ŻE...**



CHCESZ

- Jak **najlepiej** wykorzystać czas na naukę
- Dobrze płatną **pracę** w IT
- Uporządkować wiedzę
- Przechodzić rozmowy techniczne



DZISIAJ DOWIESZ SIĘ

1. Z jakich kursów, tutoriali oraz książek korzystały osoby, które dostały pracę w Javie.
 - Dzięki temu nie będziesz zakłopotany ilością materiałów do przerobienia.



DZISIAJ DOWIESZ SIĘ

2. Co musisz umieć, żeby zatrudnili Cię na Java Deva (poznasz wymagania).
 - Dzięki temu sprawdzisz czego Ci jeszcze brakuje, żeby móc wysyłać CV, a potem bez problemu zdobyć pracę.



DZISIAJ DOWIESZ SIĘ

3. Jak zaplanować drogę, nawet gdy masz około 1 godziny dziennie na naukę.
 - Dzięki temu dostaniesz pracę jako Java Developer w kilka miesięcy.



5 prostych kroków, żeby zdobyć pracę jako Java Developer w kilka miesięcy nawet jeśli masz mało wolnego czasu

1. **KROK 1 PODSTAWY - 3 MIESIĄCE**
 2. **KROK 2 DOBRE PRAKTYKI - 2 MIESIĄCE**
 3. **KROK 3 RZECZY POD PROJEKT - 2 MIESIĄCE**
 4. **KROK 4 PROJEKT I WIEDZA KOMERCYJNA DO CV - 2 MIESIĄCE**
 5. **KROK 5 PROCES REKRUTACJI - 1 MIESIĄC**
- ŁĄCZNIE 10 MIESIĘCY**



ORGANIZACYJNIE

1. Do 20:00
2. Pytania pod koniec
3. Moderator
4. Slajdy z linkami



ZOSTAŃ DO KOŃCA I...

Wygraj konsultację 1:1! (wartość 125zł)



SKUPIENIE





BARTEK KALKA

- 4 lata programista Javy/Kotlina
(Comarch, Finanteq, Britenet, Allegro)
- Zaczynałem od stażu na studiach
- Od 2 lat pomagam innym zdobyć ich pierwszą pracę w Javie
- **100+ osób** w szkoleniu Junior Java Ready
- Pomogłem już setkom osób za pomocą newslettera, kursu, LinkedIna, Facebooka, podcastu, bloga, kanału YT, mentoringów.

O MNIE

5 KROKÓW



SKĄD TE KROKI?

MOJA OBSERWACJA JAKO MENTORA

- ☑ [01/15] Wstęp do serii: od ZERA do pierwszej pracy w Javie!
- ☑ [02/15] Jesteś na poziomie pierwszym czy drugim?
- ☑ [03/15] Wszystko, co musisz wiedzieć o Gicie, Mavenie, Githubie i IntelliJ
- ☑ [04/15] Wszystko co musisz wiedzieć o Testach Jednostkowych i TDD 🔄
- ☑ [05/15] Wszystko co musisz wiedzieć o Clean Code i dobrych praktykach
- ☑ [06/15] Wszystko co musisz wiedzieć o zaawansowanej Javie ?
- ☑ [07/15] Wszystko co musisz wiedzieć o Springu i Spring Boocie ?
- ☑ [08/15] Wszystko co musisz wiedzieć o pakietowaniu i architekturze aplik
- ☑ [09/15] Wszystko, co musisz wiedzieć o bazach danych i SQL ?
- ☑ [10/15] Wszystko, co musisz wiedzieć o wystawianiu aplikacji do publicz
- ☑ [11/15] Wszystko, co musisz wiedzieć o Twoim Porządnym Projekcie Do
- ☑ [12/15] Wszystko, co musisz wiedzieć o Frontendzie jako Junior Java Dev
- ☑ [13/15] Jak się wyróżnić spośród wielu osób na podobnym poziomie? ?
- ☑ [14/15] Jak zrobić CV na które będziesz dostawać więcej odpowiedzi od f
- ☑ [15/15] Co musisz wiedzieć o procesie rekrutacyjnym na JJD? 🙌



SKĄD TE KROKI?

PODCASTY




#35 - Jak został Java Developerem? Wywiad z Szymonem Pawlikowskim
Linki z tego odcinka znajdziesz na javaready.pl/blog/o35 ---- Bartłomiej Kalka | javaready.pl

▶ Sep 16 · 1 hr 2 min




#34 - Jak został Java Developerem? Wywiad z Dymitrem Misiejukiem
Linki z tego odcinka znajdziesz na javaready.pl/blog/o34 ---- Bartłomiej Kalka | javaready.pl

▶ Aug 25 · 1 hr 2 min



#9 - Jak został Java Developerem? Wywiad z Radkiem Fedorem.
Linki wspomniane w odcinku: javaready.pl/blog/o9 ---- Bartłomiej Kalka | javaready.pl

▶ Oct 2021 · 3 min 25 sec left




#30 - Jak został Java Developerem? Wywiad z Marcinem Uzarkiem
Linki wspomniane w odcinku: javaready.pl/blog/o30 ---- Bartłomiej Kalka | javaready.pl

▶ Jul 20 · 4 min 44 sec left




#10 - Jak został Java Developerem? Wywiad z Marcinem Böhmem.
Linki na javaready.pl/blog/o10 Bartłomiej Kalka

▶ Feb 2 · 1 min 33 sec left



#28 - Jak został Java Developerem? Wywiad z Kamilem Paprockim.
Linki wspomniane w odcinku: javaready.pl/blog/o28 ---- Bartłomiej Kalka | javaready.pl

▶ Jul 13 · 54 min 55 sec



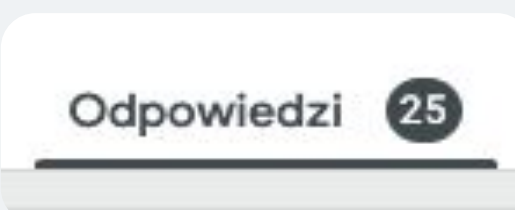
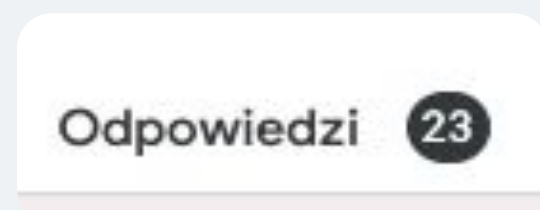
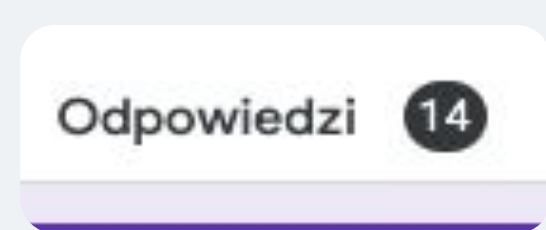
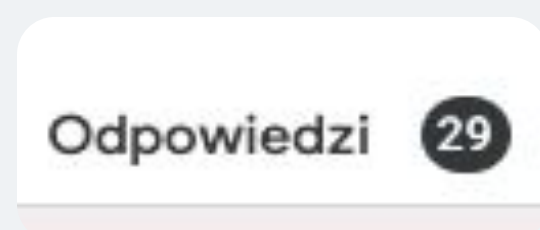
#8 - Jak został Java Developerem? Wywiad z Bartkiem Drobczykiem.
Linki wspomniane w odcinku: javaready.pl/blog/o8 ---- Bartłomiej Kalka | javaready.pl

▶ Oct 2021 · 55 min 43 sec left






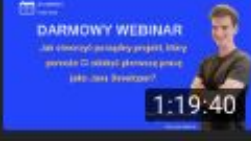


SKĄD TE KROKI?

ANKIETY



SKĄD TE KROKI?

WEBINARY

Transmisja na żywo	Typ	Widoczność	Ograniczenia	Data	Wyświe... ↓	
	Jak Zdobyć Pierwszą Pracę W Javie Bez Kom... 00:00 Wstęp, O mnie 02:05 Co robiłem źle na początku? 08:44 Jak konkurować z innymi o...	Oprogramowanie do transmisji na żywo	Publiczny	Brak	7 gru 2020 Zakończona	1 153
	10 błędów, przez które nie możesz zdobyć pie... 00:00 O mnie 03:28 Skąd te błędy? 04:45 Ucz się w jednej dziedzinie, póki nie znajdziesz w niej...	Oprogramowanie do transmisji na żywo	Publiczny	Brak	22 cze 2021 Zakończona	965
	10 rzeczy, które musisz robić, aby firmy IT odp... Spis Treści: 00:00 Start? 03:01 - Kim jestem? 05:20 - #1 Naucz się pisać testy 07:35 - #2 Stwó...	Oprogramowanie do transmisji na żywo	Publiczny	Brak	20 wrz 2021 Zakończona	901
	Jak stworzyć projekt, który pomoże Ci zdobyć... Plik ze slajdami i infografiką odbierz tutaj: https://javaready.pl/plik Zapraszam do...	Oprogramowanie do transmisji na żywo	Publiczny	Brak	20 cze 2022 Zakończona	657
	Poznaj 11 Najważniejszych Wskazówek, Dzie... Spis Treści: 00:00 Kim jestem i o czym jest ten webinar? 07:05 #1 Początki są trudne 12:05 #2...	Oprogramowanie do transmisji na żywo	Publiczny	Brak	15 mar 2022 Zakończona	392
	Jak stworzyć projekt, jak włączyć Gita, a pote... Zauważyłem, że sporo osób może mieć problem ze zrozumieniem, po co w ogóle jest Git i jak on...	Oprogramowanie do transmisji na żywo	Publiczny	Brak	13 lip 2022 Zakończona	262



ZAKŁADAMY

1 GODZINA DZIENNIE

30 dni * 1h = ~ 30h / MIESIĄC



NIE MA CUDÓW

- ciężka praca
- wytrwałość
- mądre działanie



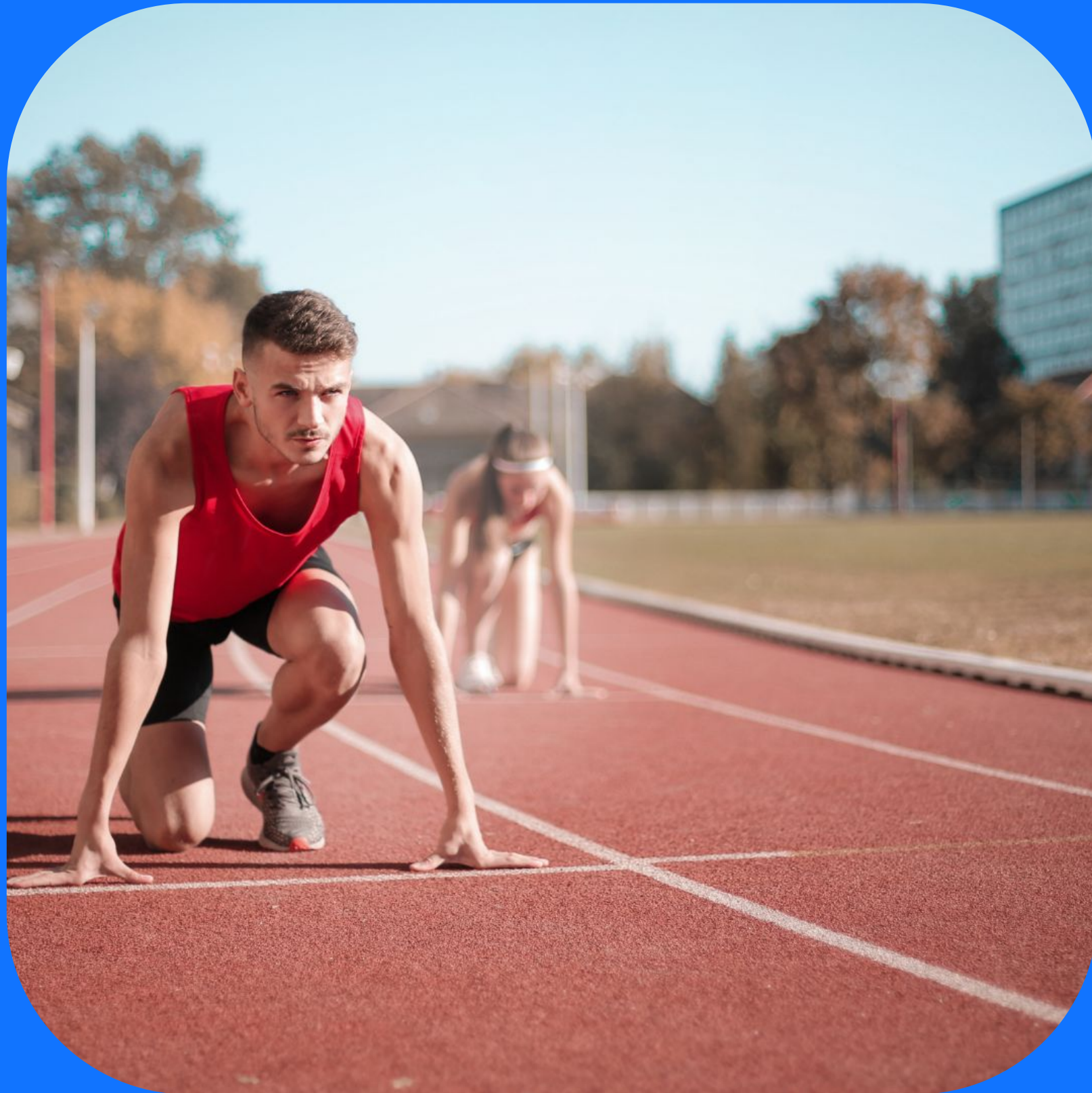
Przedstawię Ci
aktualnie
najlepszą drogę
moim zdaniem



Pro tipy

- [drzemki energetyczne](#)
- [pomodoro](#)
- wdrożyć [Pracę Głębką](#)
- prowadź notatki w GDocs
- Nie wychodź z flow jeśli nie musisz
- skup się na praktyce w 70% a nie teorii
(motywacja, lepsze postępy)





#01

PODSTAWY

KROK 1: Podstawy

3 miesiące około 90h



PS jeśli zaczniesz od października 2022 to koniec to sierpień 2023



#01

PODSTAWY

1/9 Dołączasz do darmowej serii maili

Od zera do Java Developera na javaready.pl

- ostatnio Dawid po 3 miesiącach zdobył pracę (miał podstawy C++)
- mailing pomógł mu ułożyć sobie wszystko i załatać dziury w wiedzy
- wysyłał 30 CVek

Wszystko Co Musisz Wiedzieć Żeby Zatrudnili Cię Na Junior Java Developera

- Wyróżnij się z ogromnej liczby osób, które chcą zdobyć pracę w Javie.
- Dowiesz się, na jakim poziomie musisz poznać technologie, żeby zacząć się zgłaszać do pracy. Dzięki temu **nie będziesz spalać szans u firm.**
- **W mailingu:** Git, Docker, Java, Testy Jednostkowe, TDD, Maven, Wzorce Projektowe, Architektura, Frontend, Bazy Danych, Spring/Spring Boot oraz **wszystko, co musisz wiedzieć o rekrutacjach i codziennej pracy Java Developera.**

DOŁĄCZAM DO MAILINGU

Autor



Bartek Kalka

Powiem Ci jak zdobyć pracę jako Junior Java/Kotlin Developer. Mam 4-letnie doświadczenie jako Java Developer oraz od 2 lat mentoruję innych.

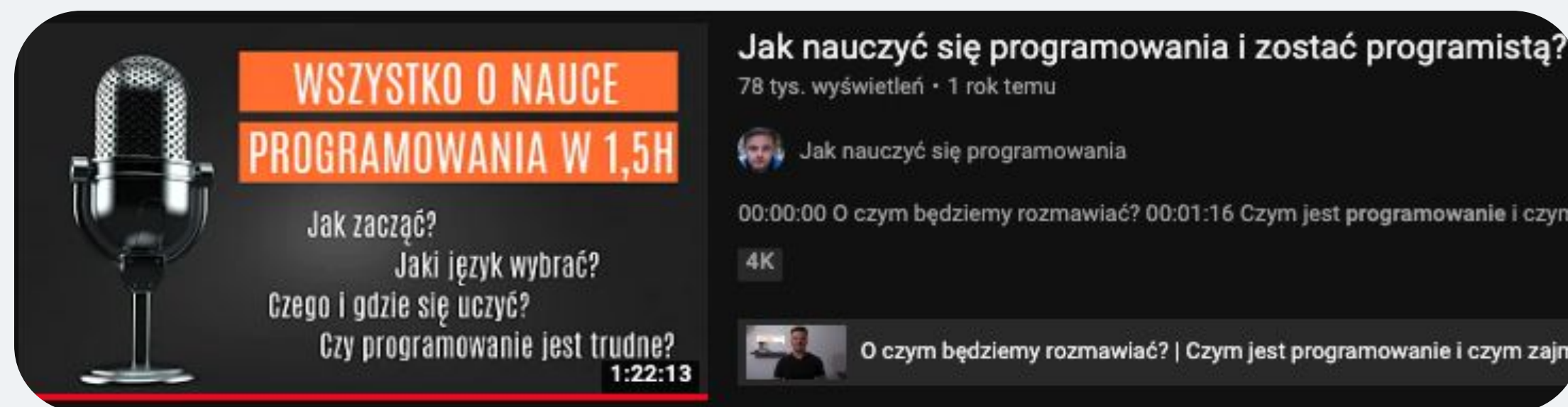
Pomogłem zdobyć pracę w Javie już **setkom osób**. Serdecznie Cię zapraszam do tego darmowego newslettera.

#01

PODSTAWY

2/9 Oglądasz video (trwa 1h 22m)

[Odcinek „Jak nauczyć się programowania i zostać programistą?” na Youtube](#)



#01

PODSTAWY

3/9 Oglądasz video (trwa 2h 30m)

[Odcinek „Java – podstawy w 2h” na Youtube](#)



#01

PODSTAWY

4/9 Inwestujesz i przerabiasz javastart

[Kurs Java Podstawy od JavaStart](#)



Kurs Java od 129 zł

Ten kurs to idealny wybór dla wszystkich osób, które dopiero rozpoczynają swoją przygodę z programowaniem i nigdy wcześniej nie miały z tym styczności. Poprowadzimy Cię od absolutnych podstaw takich jak poprawna instalacja i konfiguracja Javy, napisanie i uruchomienie pierwszego programu, aż po bardziej skomplikowane elementy związane z wykorzystaniem zewnętrznych bibliotek.

[ZOBACZ WIĘCEJ →](#)

Alternatywa to Udemy Tim Buchalka po angielsku

#01

PODSTAWY

5/9 Pobierasz aplikację mobilną Enki vs SoloLearn vs CodeGym

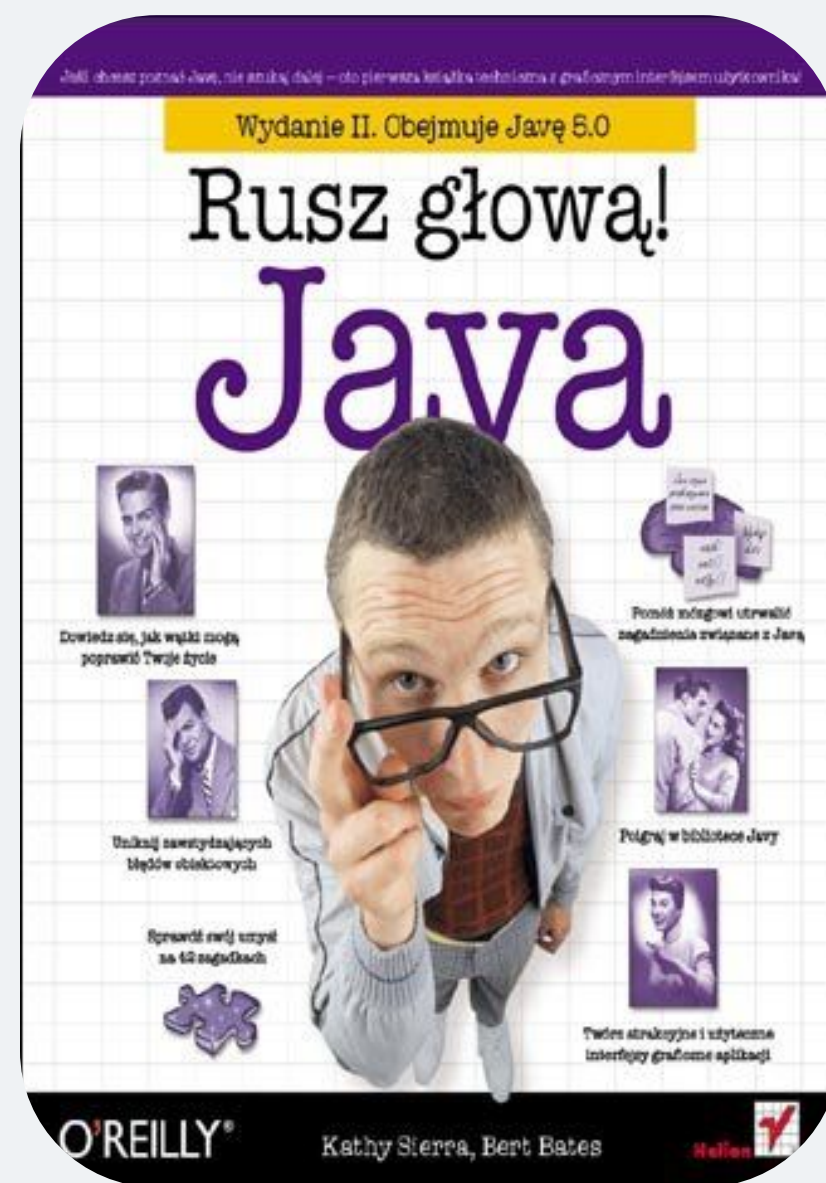


Korzystałem z Enki. Polecam CodeGym.
Potestuj wersje darmowe.

#01

PODSTAWY

6/9 Czytasz Książkę Java Rusz Głową



#01

PODSTAWY

7/9 Robisz zadania z rozwiązaniami na [w3schools](https://www.w3schools.com/java/exercises/)

of Java Exercises:

- [Basic Exercises Part-I \[150 Exercises with Solution \]](#)
- [Basic Exercises Part-II \[99 Exercises with Solution \]](#)
- [Data Types Exercises \[15 Exercises with Solution \]](#)
- [Conditional Statement Exercises \[32 Exercises with Solution \]](#)
- [Array \[77 Exercises with Solution \]](#)
- [String \[107 Exercises with Solution \]](#)
- [Date Time \[44 Exercises with Solution \]](#)
- [Methods \[23 Exercises with Solution \]](#)
- [Numbers \[28 Exercises with Solution \]](#)
- [Input-Output-File-System \[18 Exercises with Solution \]](#)
- [Collection \[126 Exercises with Solution \]](#)
- [Math \[27 Exercises with Solution \]](#)
- [Sorting \[19 Exercises with Solution \]](#)
- [Search \[7 Exercises with Solution \]](#)
- [Regular Expression \[30 Exercises with Solution \]](#)

More..

100% za darmo.

#01

PODSTAWY

8/9 Inwestujesz ~ 20 PLN/m i robisz CodeGym!

Będzie przeplatać się z krokiem 2 “Dobre Praktyki” i super!



Składnia Java

Misja „Składnia Java” jest odpowiednia nawet dla osób, które nigdy wcześniej niczego nie programowały. Poznasz dzięki niej klasy, obiekty, metody, zmienne, typy danych, tablice, operatory warunkowe i pętle. Nauczysz się podstaw kolekcji i programowania obiektowego, a także zaczniesz pracować w IntelliJ IDEA – środowisku programistycznym używanym przez programistów z całego świata!



Podstawy Java

Misja „Podstawy Java” jest przeznaczona dla osób, które przeszły już podstawowe szkolenie Java (kończąc misję „Składnia Java”). Do przejścia masz 10 fascynujących poziomów, na których nauczysz się podstaw programowania obiektowego, zapoznasz się ze strumieniami, serializacją i przeciążaniem metod, a także nauczysz się dużo o interfejsach i wielokrotnym dziedziczeniu.



Kolekcje w Java

Kurs „Kolekcje w Java” jest poświęcony dogłębnej analizie kolekcji Java i nie tylko. Opowiemy o pracy z plikami i archiwami oraz o wzorcach projektowych i sposobach ich używania. Zdobędziesz doświadczenie w JSON, Guava, Apache Commons Collections oraz JUnit. Dowiesz się więcej na temat garbage collection w Javie oraz poznasz Gita, RMI, DynamicProxy i JavaScript.

Początek jest za free. Przetestuj

#01

PODSTAWY

9/9 Bardzo opcjonalnie

- Darmowe. Możesz zobaczyć po angielsku [CS50 computer science](#)
- [Kurs Javy](#) na Udemy po angielsku Java od zera od Tim buchalka (zadania)

#01

**PODSUMOWANIE:
PODSTAWY**

3 miesiące

1. Dołączasz do darmowej serii od Zera do Java Developera javaready.pl
2. Oglądasz [Odcinek](#) „Jak nauczyć się programowania i zostać programistą?”
3. Oglądasz [Odcinek](#) „Java – podstawy w 2h” na Youtube
4. Przerabiasz [javastart](#)
5. Pobierasz aplikację mobilną (polecam CodeGym)
6. Czytasz Książkę Java Rusz Głową!
7. Robisz zadania na [w3schools](#)
8. Inwestujesz i przerabiasz [CodeGym!](#)
9. Opcjonalnie:
 - a. Możesz zobaczyć po angielsku [cs50 computer science](#).
 - b. [Udemy kurs](#) od zera od Tim Buchalka (zadania praktyczne)

#02

DOBRE PRAKTYKI

KROK 2: Dobre Praktyki

2 miesiące około 60h



#02

DOBRE PRAKTYKI

1/6 W CodeGym też będziesz mieć opisane dobre praktyki

Na przykład:

Jaka jest różnica między ArrayList a LinkedList (bardzo częste pytanie na rekrutacjach).

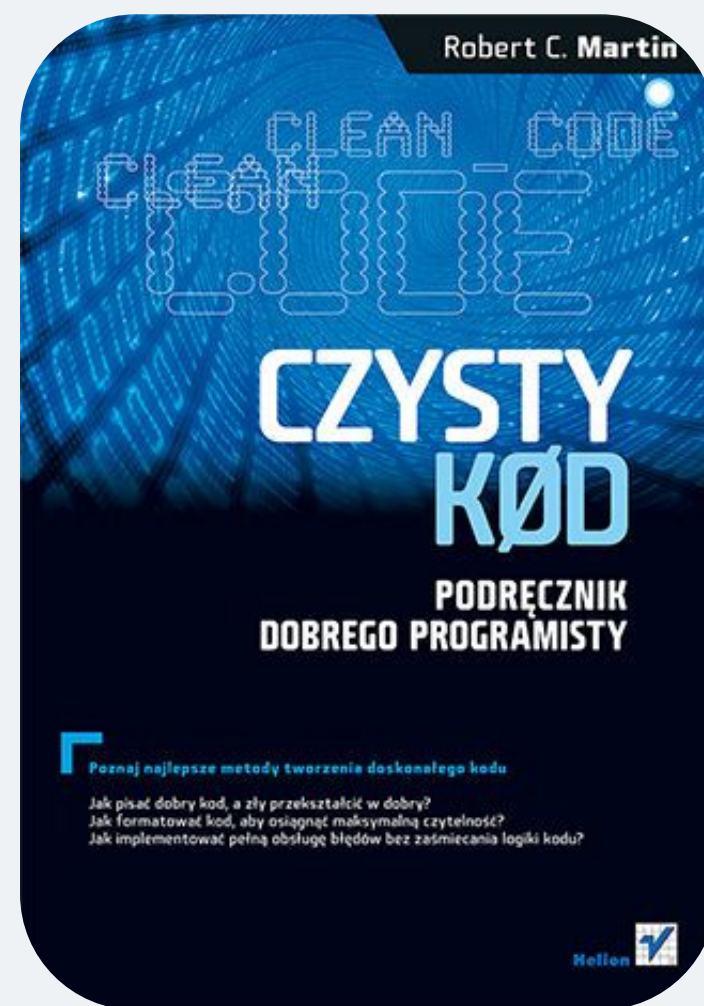
Dowiedz się tam :)

#02

DOBRE PRAKTYKI

2/6 Czytasz pierwsze 8 rozdziałów z Książki Czysty Kod

To kwintesencja dobrych praktyk pisania kodu w Javie w codziennej pracy



#02

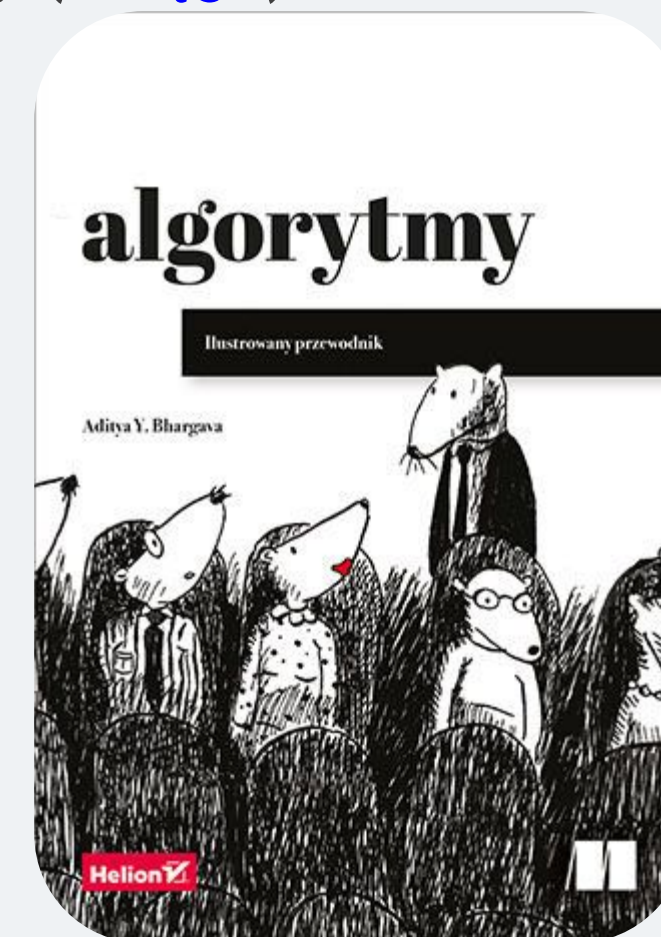
DOBRE PRAKTYKI

3/6 Czytasz Książkę: Algorytmy Ilustrowany Przewodnik Aditya Bhargava

- Pomoże Ci na rekrutacjach
- Nie byłeś na studiach i nie wiesz czym jest złożoność obliczeniowa
- Notacja dużego O = Big-O Complexity ([ściąga](#))

Data Structure	Time Complexity								Space Complexity
	Average				Worst				
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
Array	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Stack	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
Queue	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
Singly-Linked List	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
Doubly-Linked List	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
Skip List	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n \log(n))$
Hash Table	N/A	$O(1)$	$O(1)$	$O(1)$	N/A	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Binary Search Tree	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Cartesian Tree	N/A	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	N/A	$O(n)$	$O(n)$	$O(n)$	$O(n)$
B-Tree	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
Red-Black Tree	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
Splay Tree	N/A	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	N/A	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
AVL Tree	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
KD Tree	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	
Quicksort	$O(n \log(n))$	$O(n \log(n))$	$O(n^2)$	$O(\log(n))$
Mergesort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	$O(n)$
Timsort	$O(n)$	$O(n \log(n))$	$O(n \log(n))$	$O(n)$
Heapsort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	$O(1)$
Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$



#02

DOBRE PRAKTYKI

4/6 Przerabiasz Kurs Wzorce Projektowe

Przerabiasz [kurs](#) - Lepiej zrozumiesz obiektówkę, kilkanaście wzorców, kilkadziesiąt zadań i przykładów

▼ Obserwator (Observer)	5 wykładu • 22 min
▼ Singleton	6 wykładu • 16 min
▼ Builder	7 wykładu • 28 min
▼ Fabryka (Factory)	8 wykładu • 25 min
▼ Pyłek (Flyweight)	5 wykładu • 23 min
▼ Adapter	6 wykładu • 16 min
▼ Dekorator	5 wykładu • 19 min
▼ Łańcuch Odpowiedzialności (Chain of Responsibility)	5 wykładu • 24 min
▼ Metoda Szablonowa (Template Method)	5 wykładu • 12 min

#02

DOBRE PRAKTYKI

5/6 (Opcja) Czytasz artykuł o Dobrych Praktykach OOP (Object Oriented Programming)

Czytasz [artykuł](#)

5 dobrych praktyk OOP, które musisz poznać:

1. **Niemutowalność (Immutability)**

- a. prosta, ale jedna z ważniejszych praktyk w aktualnych czasach
 - i. jeśli raz stworzysz obiekt, to nie da się go nadpisać - inny wątek również nie nadpisze stanu tego obiektu. Kluczowa w aplikacjach wielowątkowych, dlatego że odpadają typowe problemy.
 - ii. żeby zmodyfikować taki obiekt, trzeba stworzyć nowy obiekt ze zmienionymi wartościami.
 - iii. ułatwia debugowanie aplikacji, dlatego, że wszystkie niemutowalne obiekty mają stan, który miały w momencie ich tworzenia

2. **KISS (Keep It Simple Stupid)**

- a. Chodzi o to, żeby nie pisać dziesięciu skomplikowanych linii kodu, który tylko Ty rozumiesz (a pewnie za kilka tygodni też będziesz mieć problemy ze zrozumieniem „co autor miał na myśli”). Lepiej zrobić 2-3 metody, które są dobrze nazwane i każdą z nich utrzymywać osobno.
- b. Jeśli coś da się zrobić prościej, to powinieneś to preferować, ze względu na innych (i siebie w przyszłości).

3. **DRY (Don't Repeat Yourself)**

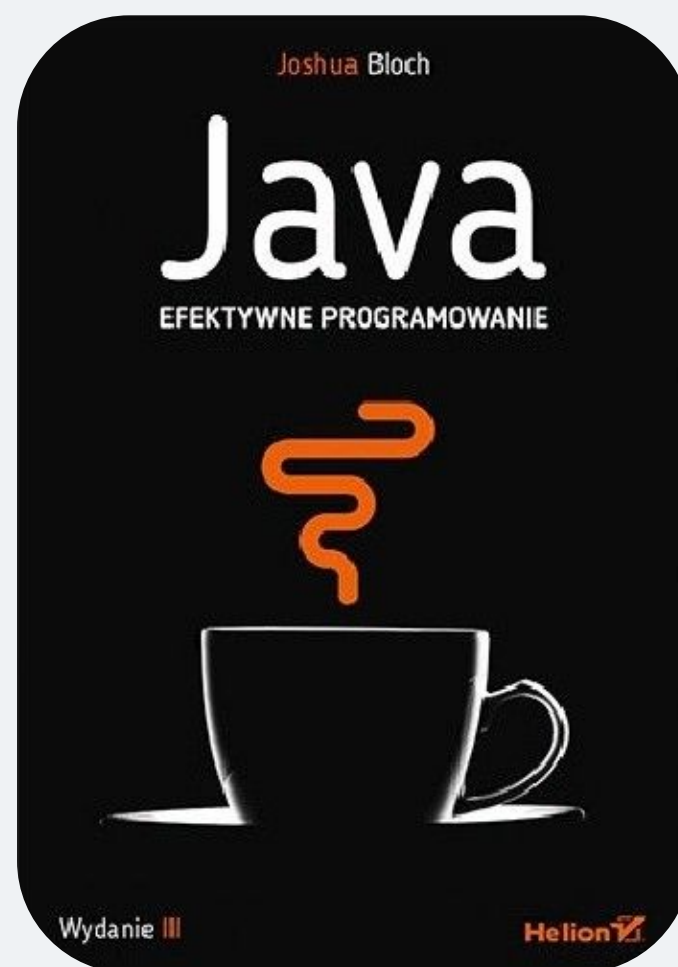
- a. Duplikowanie kodu ma swoje **wady** i **zalety**, a ta zasada nie powinna być stosowana bez przemyślenia.
- b. Duplikowanie kodu powoduje problem tego typu, że jeśli kopiowany algorytm się zmieni to trzeba go zmieniać w wielu miejscach, zamiast w jednym.
- c. Jeśli masz już w projekcie klasę, która odpowiada za wysyłanie maili, to raczej nie powinieneś kopiować metod tej klasy do swojej nowej klasy, żeby skorzystać z jej funkcjonalności. Powinieneś raczej skorzystać z istniejącego

#02

DOBRE PRAKTYKI

6/6 Czytasz Książkę Java. Efektywne programowanie Joshua Bloch

Wchodzisz w Javę zaawansowane tematy i rady od praktyków (taki kod spotkasz w pracy), ale **według tego** jak tu napisałem w [artykule](#)



Przeczytaj tematy z książki *Java Efektywne Programowanie*

3 powody, dla których warto przerobić tę książkę?

1. Autorem jest jeden z twórców Javy: Joshua Bloch - inżynier oprogramowania, obecnie pracujący w Google. Prowadził projekt i implementację licznych funkcjonalności języka i platformy Java, między innymi Java Collections.
2. Programiści korzystają z tych praktyk w codziennej pracy w Javie, są to rzeczy, z których wszyscy korzystają i wiedzą, że tak trzeba robić, ale niekoniecznie każdy wie, że jest to z tej książki ;)
3. Wiele rzeczy, o których dzisiaj wspominałem jest pokrytych w tej książce. **SIC!** Nie jestem fanem przerabiania całych książek na sucho, dlatego wskażę Ci tylko wymagane minimum, które warto poznać.

Przejdźmy do...

18 tematów, które musisz poznać:

Temat 3: Wymuszanie właściwości Singleton za pomocą prywatnego konstruktora lub typu enum

Dlaczego warto poznać ten temat?

Warto wiedzieć czym jest Singleton, ponieważ w pracy spotkasz się z tym bardzo często - a na pewno na Twojej rozmowie rekrutacyjnej padnie pytanie o ten wzorzec

Temat 5: Wstrzykiwanie zależności

Dlaczego warto poznać ten temat?

Jak widzisz nie tylko w Springu jest Wstrzykiwanie Zależności. Możesz to uzyskać w Javie. Poznaj ten mechanizm, a potem będzie Ci łatwiej zrozumieć Springa. Zalet wstrzykiwania jest mnóstwo. Na przykład łatwiejsze testowanie aplikacji czy zmniejszanie zależności między klasami.

Temat 9: Try z zasobami (try with resources)

#02

PODSUMOWANIE:
DOBRE PRAKTYKI

2 miesiące

1. W CodeGym też będziesz mieć opisane dobre praktyki
2. Czytasz pierwsze 8 rozdziałów z Książki Czysty Kod
3. Czytasz Książkę Algorytmy Ilustrowany przewodnik Aditya Bharga
4. Przerabiasz Kurs Wzorce Projektowe
5. (Opcja) Czytasz [artykuł](#) o Dobrych Praktykach OOP
6. Czytasz Książkę Java. Efektywne programowanie od Joshua Bloch

#03

NAUKA POD PROJEKT

KROK 3: Nauka Pod Projekt

2 miesiące około 60h



#03

NAUKA POD PROJEKT

1/5 Poznajesz Git, Github, Maven/Gradle, IntelliJ

1. Oglądasz video: [Jak stworzyć projekt, jak włączyć Gita, a potem wrzucić do Githuba tylko za pomocą IntelliJ?](#)
2. Czytasz darmowy [artykuł](#)

Miałem teraz czas obejrzeć Twoje video o Git i Github od początku do końca. [Miesiąc temu robiłem 17 godzinny!!!!](#) tak 17 godzinny!!!! kurs na Udemy o git i github, zrezygnowałem po 3 godzinach bo coś podejrzewałem, że to nie o to chodzi (oczywiście po tych 3 godzinach dalej nie umiałem używać ani Gita ani Githuba miałem za to dużo zupełnie zbędnej wiedzy i informacji.

[Twoje video 1 godzina i wszystko umiem używać i wiem o co chodzi](#) (w zasadzie mniej niż godzina bo duża część była o Maven). To jest właśnie różnica pomiędzy uczeniem, a udawaniem, że się uczy. Kudos Bartek!!! w zasadzie nic dodać nic ująć, brawa, kurtyna, aktorzy schodzą ze sceny, widownia/uczniowie zachwyceni ;-)



#03

NAUKA POD
PROJEKT

2/5 Poznajesz Testy Jednostkowe

1. Przerabiasz [Kurs](#) Testy Jednostkowe (55 zł)
2. (Opcja) Czytasz [Ebook](#): Złe testy, Dobre testy (90 zł)
3. Czytasz darmowy [artykuł](#) o testach jednostkowych i TDD

▼ JUnit 5	18 wykładu • 2 godz. 23 min
▼ Dobre praktyki oraz konwencje tworzenia testów	5 wykładu • 32 min
▼ Mockito 2	11 wykładu • 1 godz. 50 min
▼ Test Driven Development	6 wykładu • 32 min

#03

NAUKA POD
PROJEKT

3/5 Poznajesz Springa i Spring Boota

1. Przerabiasz [darmowy tutorial](#) Spring Boot, REST, Hibernate
2. Przerabiasz Darmowe [Oficjalne Tutoriale](#) od Spring Boota
3. Przerabiasz Książkę [Spring Start Here](#) (przykłady kodu do pobrania, lepsza niż Spring w Akcji tylko, że po angielsku)
4. (Opcja) Czytasz książkę Spring w Akcji (do snu)
5. Przerabiasz [Kurs](#) Javastart Webowe
6. Przerabiasz [Kurs](#) JavaStart Spring
7. (Opcja) Przerabiasz [Kurs](#) Springa po polsku
8. Przerabiasz [Kurs](#) Springa po polsku (nie od podstaw)
9. (Opcja) Przerabiasz [Kurs](#) Springa po angielsku od Chad Darby

#03

NAUKA POD PROJEKT

4/5 Poznajesz bazy danych i Hibernate

1. Przerabiasz [Kurs](#) Hibernate
2. Czytasz darmowy [artykuł](#) o SQL i NoSQL
3. (Opcja) Przerabiasz [książkę](#) SQL teoria i praktyka

Poznaj pojęcia bazodanowe i bazy danych.

Najważniejszym z tego wpisu jest **SQL**.

To on jest najczęściej spotykany na rozmowach rekrutacyjnych i w zadaniach rekrutacyjnych.

Oprócz tego warto wiedzieć i znać, z czym będziesz mieć do czynienia w **codziennej pracy z bazami jako programista**.

Postaram się wypisać **niezbędne minimum**, podać wartościowe materiały i dać **zarys praktyczny wszystkiego**, co warto wiedzieć o bazach danych jako **entry level** programista Javy.

Wiele zależy od tego co firma, do której się rekrutujesz **wykorzystuje**.

1. Czy jest to baza **SQL**?
2. A może baza **NoSQL**?

Taką informację łatwo zdobyć z **opisu oferty pracy**.

Zobacz poniżej.

Technologie, których używamy

Wymagane

Java

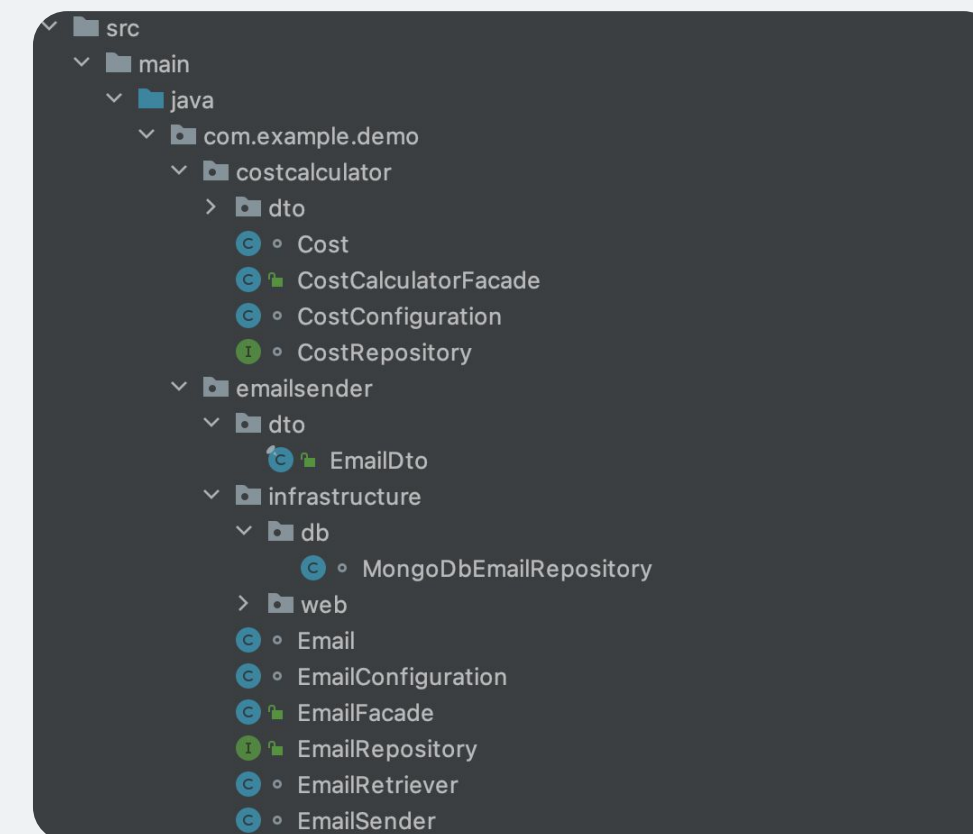
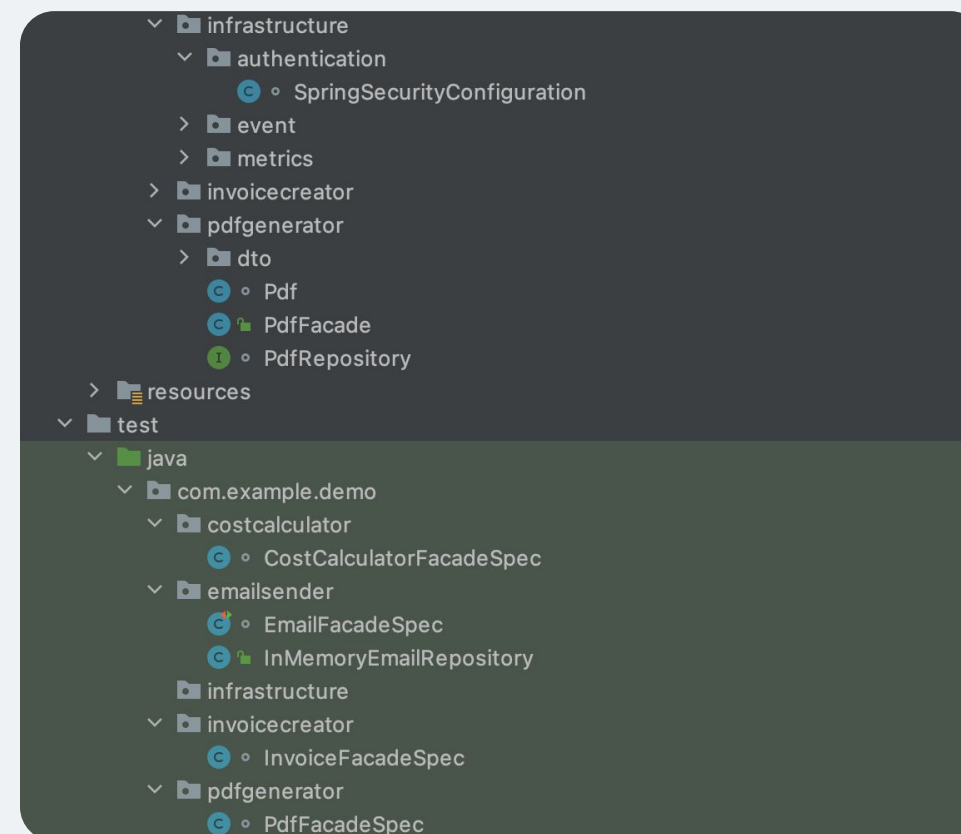
SQL

#03

NAUKA POD PROJEKT

5/5 Poznajesz architekturę i pakietowanie

1. Przerabiasz [Warsztat](#) Architektura Warstwowa
2. Czytasz darmowy [artykuł](#) - podstawy architektury



#03

PODSUMOWANIE:

NAUKA POD

PROJEKT

2 miesiące

1. Poznajesz Git, Github, Maven/Gradle, IntelliJ
2. Poznajesz Testy Jednostkowe
3. Poznajesz Springa i Spring Boota
4. Poznajesz bazy danych i Hibernate
5. Poznajesz architekturę i pakietowanie

#04

**PROJEKT i
WIEDZA KOMERCYJNA**

KROK 4: Projekt i Wiedza Komercyjna

2 miesiące około 60h

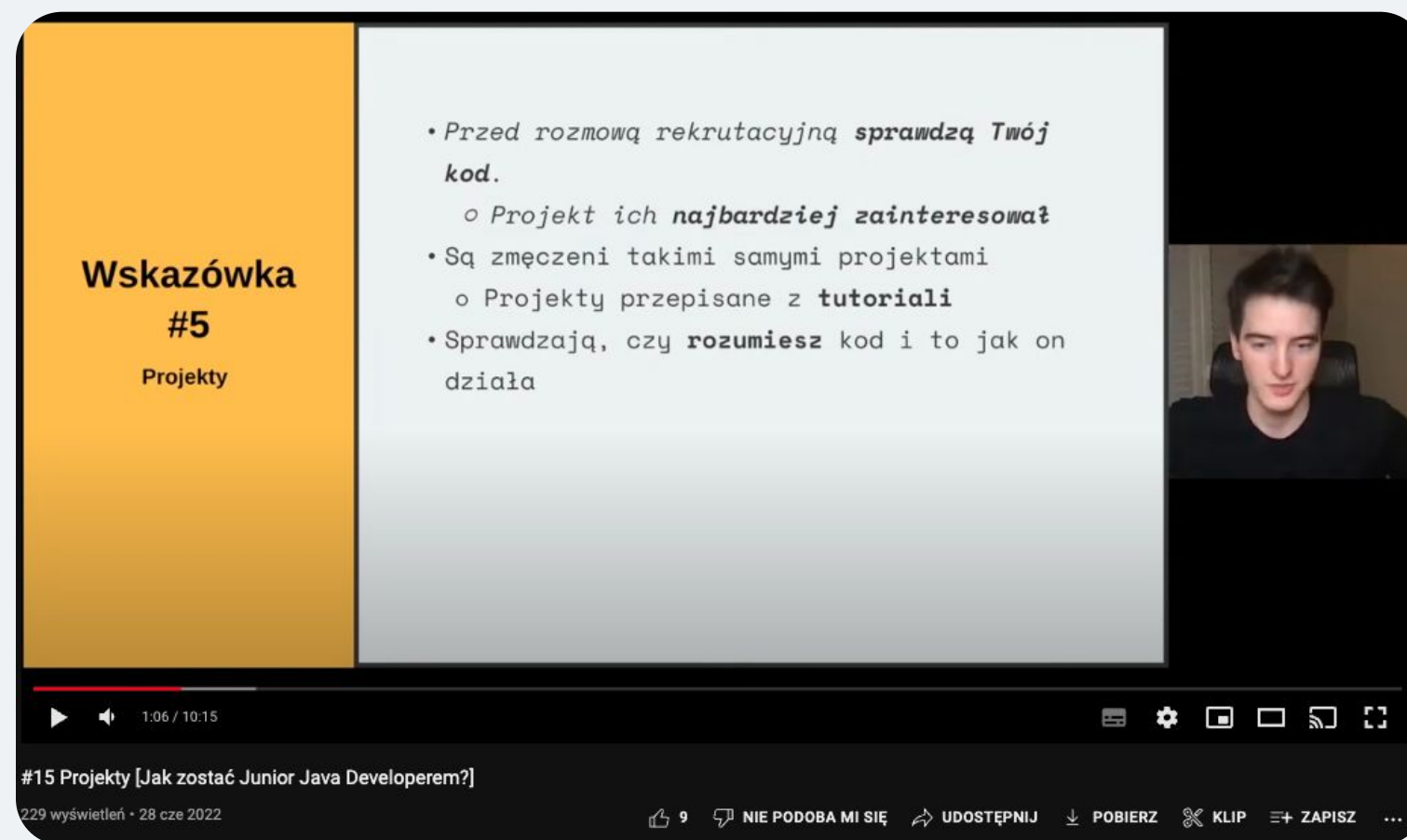


#04

PROJEKT I WIEDZA
KOMERCYJNA

1/6 Oglądasz video

1. [Dlaczego **nie warto** robić CRUDa do CV](#)
2. Jeśli chcesz CRUDA, to zrób potem drugi projekt



#04

PROJEKT I WIEDZA
KOMERCYJNA

2/6 Podpatrujesz projekty osób, które zdobyły pracę w Javie

1. [PROJEKT 1](#)
2. [PROJEKT 2](#)
3. [PROJEKT 3 \(bookingFront & bookingBackend\)](#)
4. [PROJEKT 4](#)
5. [PROJEKT 5](#)

#04

PROJEKT I WIEDZA
KOMERCYJNA

3/6 Oglądasz video

[Jak stworzyć projekt, który pomoże Ci zdobyć pierwszą pracę jako Java Dev?](#)

00:07:10 - Krok 1 Określ temat projektu

00:08:01 - Krok 2 Technologie z 10 ofert

00:09:37 - Krok 3 Zbierz i spisz wymagania

00:13:02 - Krok 4 Wypisz scenariusze użycia

00:16:24 - Krok 5 Architektura systemu

00:18:57 - Krok 6 Podział na moduły

00:22:00 - Krok 7 Zrób Diagram

00:29:09 - Krok 8 Stwórz projekt w IDE i ustaw go

00:30:22 - Krok 9 Architektura aplikacji i pakietowanie

00:38:12 - Krok 10 Napisz Testy Jednostkowe

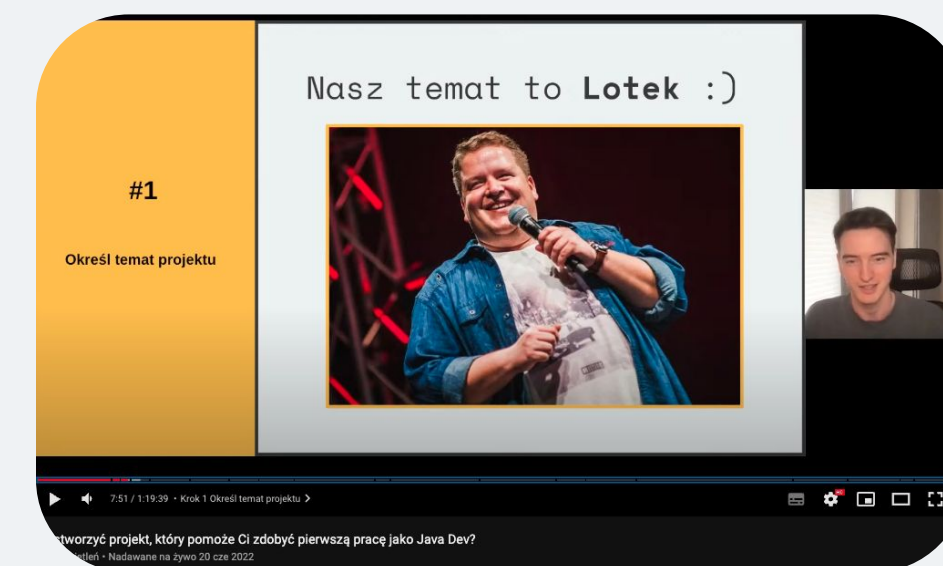
00:44:55 - Krok 11 Korekta wymagań

00:47:30 - Krok 12 Podepnij Springa i Spring Boota

00:51:55 - Krok 13 Napisz Testy Integracyjne

01:05:49 - Krok 14 Docker

01:10:30 - Krok 15 AWS HEROKU domena .pl



#04

PROJEKT I WIEDZA
KOMERCYJNA

4/6 (Opcja) Czytasz Wpis

Wybierasz temat projektu i technologie pod pierwszą pracę

2. Stwórz projekt wartościowy dla Ciebie i pracodawców

Jeśli nie masz czym się pochwalić to marne szanse, że uda Ci się przebić wśród wielu **WIEEELUUU** kandydatów na stanowisko Junior Java Developera.

Każda osoba startująca na to stanowisko musi mieć swój projekt.

Koniec kropka.



Ale niby jaki projekt jest wartościowy? Co takiego mam stworzyć?

#04

PROJEKT I WIEDZA
KOMERCYJNA

5/6 Poznajesz i stosujesz w projekcie wiedzę komercyjną

1. Docker
2. TestContainers
3. Wiremock
4. HTTP clients
5. Testowanie w pamięci
6. Testy jednostkowe i integracyjne
7. Kod jaki spotkasz w pracy

#04

PROJEKT I WIEDZA
KOMERCYJNA

6/6 Szukasz Feedbacku/CodeReview

1. Oglądasz video [11 Najważniejszych Wskazówek, Dzięki Którym Znajdziesz Pierwszą Pracę w Javie](#)
2. Szukasz grupę wsparcia i wymieniającą doświadczenia
3. Szukasz Mentora
4. Feedback Twoich aktualnych umiejętności
5. Douczasz się
6. 2 projekty z Code Review
7. 1 projekt z Code Review do CV

#04

PODSUMOWANIE:
PROJEKT I WIEDZA
KOMERCYJNA

2 miesiące

1. Oglądasz [Dlaczego nie warto robić CRUDA](#)
2. Podpatrujesz projekty osób, które zdobyły pracę w Javie
3. Oglądasz [Jak stworzyć projekt, który pomoże Ci zdobyć pierwszą pracę jako Java Dev?](#)
4. (Opcja) Czytasz [Wpis](#) (temat i tech)
5. Poznajesz i stosujesz w projekcie wiedzę komercyjną
6. Szukasz Feedbacku/CodeReview

#05

REKRUTACJE

KROK 5: Rekrutacje

1 miesiąc około 30h



#05

REKRUTACJE

1/8 Rekrutacje = nie wszystko zależy od Ciebie

- angielski w firmie
- programiści w firmie
- liczba ofert
- twoja dostępność
- typ oferty
- rekruterzy
- procesy rekrutacyjne

#05

REKRUTACJE

2/8 Poznajesz codzienną pracę programisty

1. Scrum
2. Sprint
3. Daily
4. Story Pointy
5. Jira
6. Github/Gitlab
7. Jenkins Ci&Cd
8. Codereview
9. Pull Request



Moduł 2 Wiedza Komercyjna – Teoria [58 minut]



- Lekcja 1: Powitanie.
- Lekcja 2: Jak ten moduł wyróżni względem innych kandydatów?
- Lekcja 3: Scrum.
- Lekcja 4: Sprint.
- Lekcja 5: Daily.
- Lekcja 6: Review.
- Lekcja 7: Retrospektywa.
- Lekcja 8: Refinement i Story Pointy.
- Lekcja 9: Planning.
- Lekcja 10: Praca Domowa 🏠
- Lekcja 11: Gitlab.
- Lekcja 12: Code Review i Merge Requesty
- Lekcja 13: JIRA.
- Lekcja 14: Jenkins.
- Lekcja 15: Praca Domowa 🏠

#05

REKRUTACJE

3/8 Oglądasz video

[10 rzeczy, które musisz zrobić, aby firmy IT odpowiadały na Twoje CV Junior Java Developera](#)



#05

REKRUTACJE

4/8 Tworzysz CV i sprawdzasz innych

1. Skorzystaj z tego [szablону CV](#) i wypełnij
2. Sprawdź CV osób, które dostały prace:
 - a. [CV 1](#)
 - b. [CV 2](#)
 - c. [CV 3](#)
 - d. [CV 4](#)
3. Dajesz swoje CV do oceny (Warto wysłać?)

#05

REKRUTACJE

5/8 Wysyłasz CV

1. Wysyłasz 30-50 CV i patrzysz co się dzieje
2. [300 firm](#) do pracy dla Java Developera
3. Proś o feedback
 - a. czy iść dalej, czy się cofnąć

#05

REKRUTACJE

6/8 Szkolisz angielski

1. angielski na rozmowach
 - a. preply.com – nauka angielskiego z nativem
 - b. etutor.pl - codziennie robisz
2. Staraj się książki, kursy i dokumentacje przerabiać po angielsku

#05

REKRUTACJE

7/8 Oglądasz video

Zadania i Rozmowy rekrutacyjne

14:33 / 20:23 • Krok #9 Rozmowy Rekrutacyjne >

Jak zdobyć swoją pierwszą pracę jako Java Developer? Część 4/4.

241 wyświetleń • 23 cze 2022

13 NIE PODOBA MI SIĘ UDOSTĘPNIJ POBIERZ KLIP ZAPISZ ...

#05

REKRUTACJE

8/8 Robisz postępy codziennie

1. Rób powtórki, utrwalaj wiedzę
2. Rozwiązuj codziennie zadania programistyczno-logiczne [DailyCodingProblem](#)
3. Najczęstsze pytania rekrutacyjne
4. Zadania pod rekrutacje
5. Dostajesz z 10 zaproszeń na rozmowy
6. Pytaj o feedback po rozmowach!

#05

PODSUMOWANIE:
REKRUTACJE

1 miesiąc

1. Rekrutacje = nie wszystko zależy od Ciebie
2. Poznajesz codzienną pracę programisty
3. Oglądasz video [10 rzeczy, które musisz robić, aby firmy IT odpowiadały na Twoje CV Junior Java Developera](#)
4. Tworzysz CV i sprawdzasz innych
5. Wysyłasz CV
6. Szkolisz angielski
7. Oglądasz video [Zadania i Rozmowy rekrutacyjne](#)
8. Robisz postępy codziennie

KROK 6: Koniec!



1. PODSTAWY - 3 MIESIĄCE
 2. DOBRE PRAKTYKI - 2 MIESIĄCE
 3. RZECZY POD PROJEKT - 2 MSC.
 4. PROJEKT I W. KOM. - 2 MSC.
 5. REKRUTACJE - 1 MIESIĄC
- ŁĄCZNIE ~ 10 MIESIĘCY ~ 300h

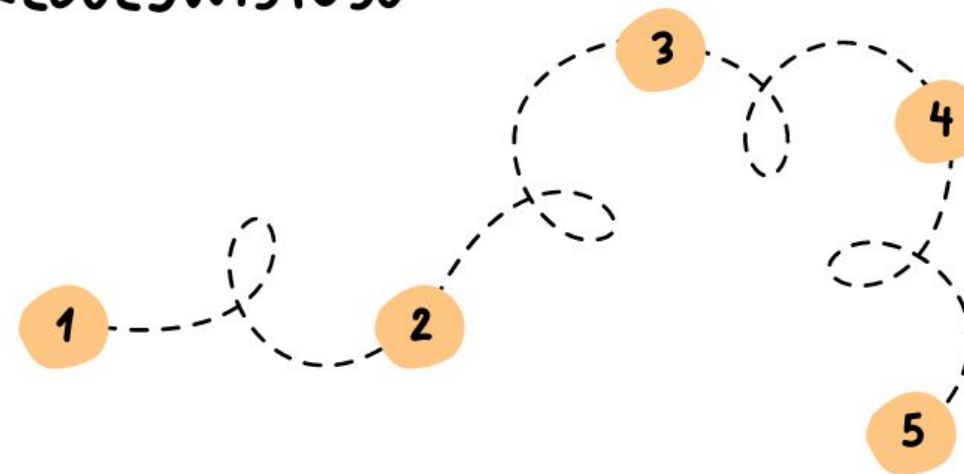


NIE PRZEJMUY SIĘ!

OCZEKIWANIA



RZECZYWISTOŚĆ





www.JavaReady.pl/plik



 Add cover  Add comment

Materiały z webinarów | javaready.pl

Rozwiń interesujący Cię webinar, a znajdziesz odpowiednie materiały.

- Webinar (20.09.2022): 5 prostych kroków, żeby zdobyć pracę jako Java Developer w kilka miesięcy nawet jeśli masz mało wolnego czasu?