



# Teoria programowania

//Paulina Kucharczyk

## **01.** Język

Definicja i sposoby zapisu

## **03.** Architektura

Wzorce projektowe

## **02.** Klasyfikacja

Podział języków  
programowania

## **04.** Zadanie

Praca w grupach

01

# Język

Definicja i sposoby zapisu

# Język

“Sposób porozumiewania się ludzi  
pewnego środowiska lub zawodu oraz  
zapisu i przekazywania informacji w  
jakiejś dziedzinie wiedzy.”

~ Wikipedia

# Język programowania



“Zbiór zasad określających, kiedy ciąg symboli tworzy program komputerowy oraz jakie obliczenia opisuje.”

~ Wikipedia

W skrócie - formalny język do zapisywania algorytmów.

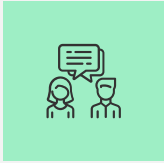
**Algorytm?**

# Algorytm



“Skończony ciąg jasno zdefiniowanych  
czynności koniecznych do wykonania  
pewnego rodzaju zadań”  
~ Wikipedia

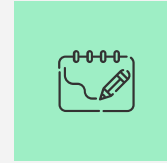
# Sposoby zapisu algorytmów



Opis słowny



Lista kroków



Schemat blokowy



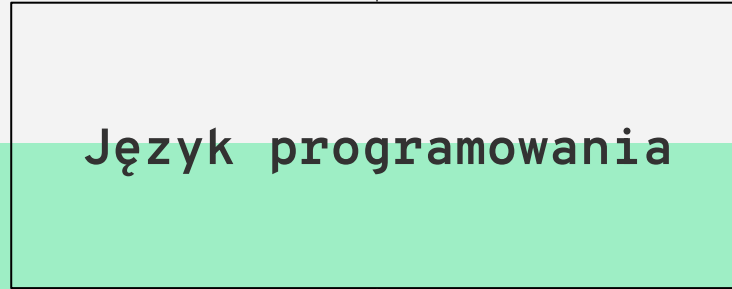
Pseudokod



Język formalny



Składnia i reguły



Język programowania



Instrukcje i słowa kluczowe

02

# Klasyfikacja

Podział języków programowania

# Podział języków programowania

01

## Paradygmaty

Podział ze względu na samodzielność komputera w rozwiązywanie problemu

02

## Generacje

Podział ze względu na zaawansowanie i sposób komunikacji

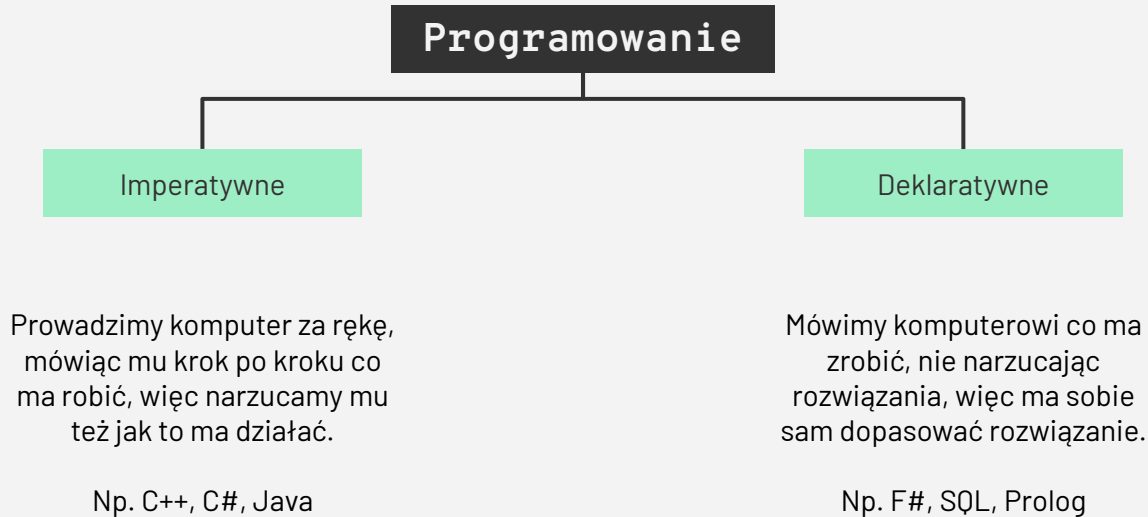
# Paradygmat języka

“Przyjęty sposób widzenia rzeczywistości  
w danej dziedzinie, doktrynie itp.”

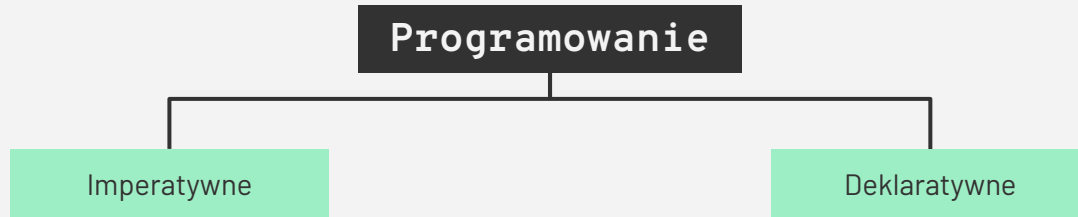
~ PWN

W skrócie - przyjęte przez wszystkich  
niepisane zasady lub przełamany  
schemat myślenia jak np. geocentryzm.

# Podział paradygmatów



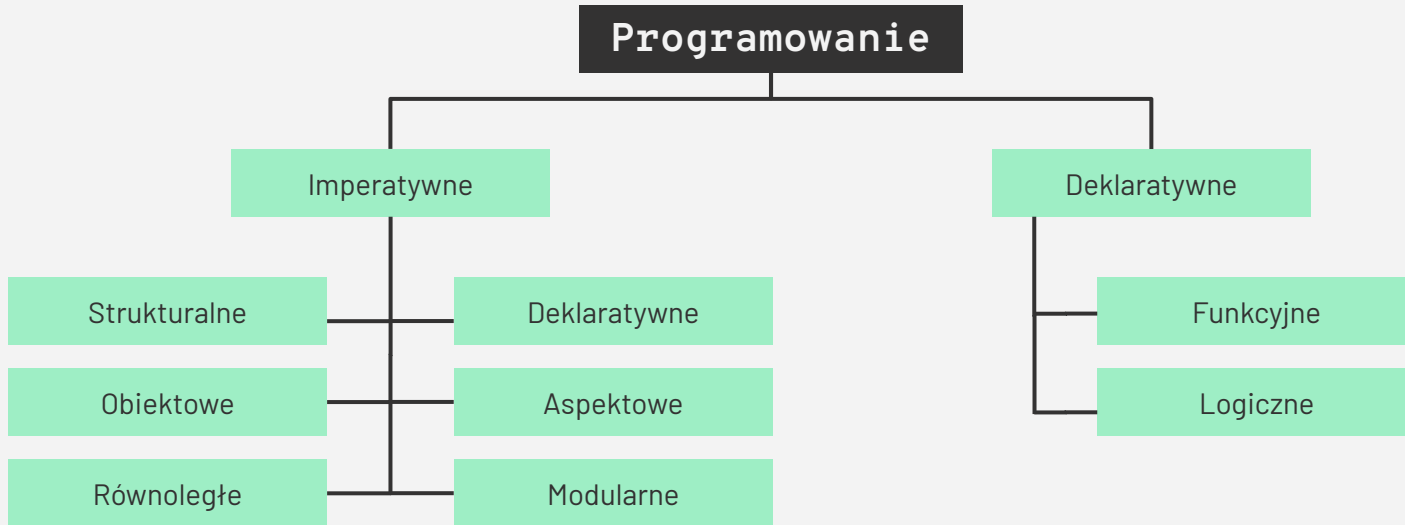
# Podział paradygmatów



- Działanie jest prostsze do zrozumienia dla początkujących.
- Łatwo śledzić w nim wszystkie kroki.
- Wymaga pisania bardziej rozbudowanego kodu. Mogą się pojawić problemy z czytelnością, zwłaszcza w przypadku większych programów.

- Kod jest bardziej prosty i czytelny.
- Pozwala skupić się na wyniku a nie na sposobie w jaki rozwiązujemy problem.
- Wymaga większego doświadczenia i dobrej znajomości mechanizmów języka, w którym pracujemy.

# Podział paradygmatów



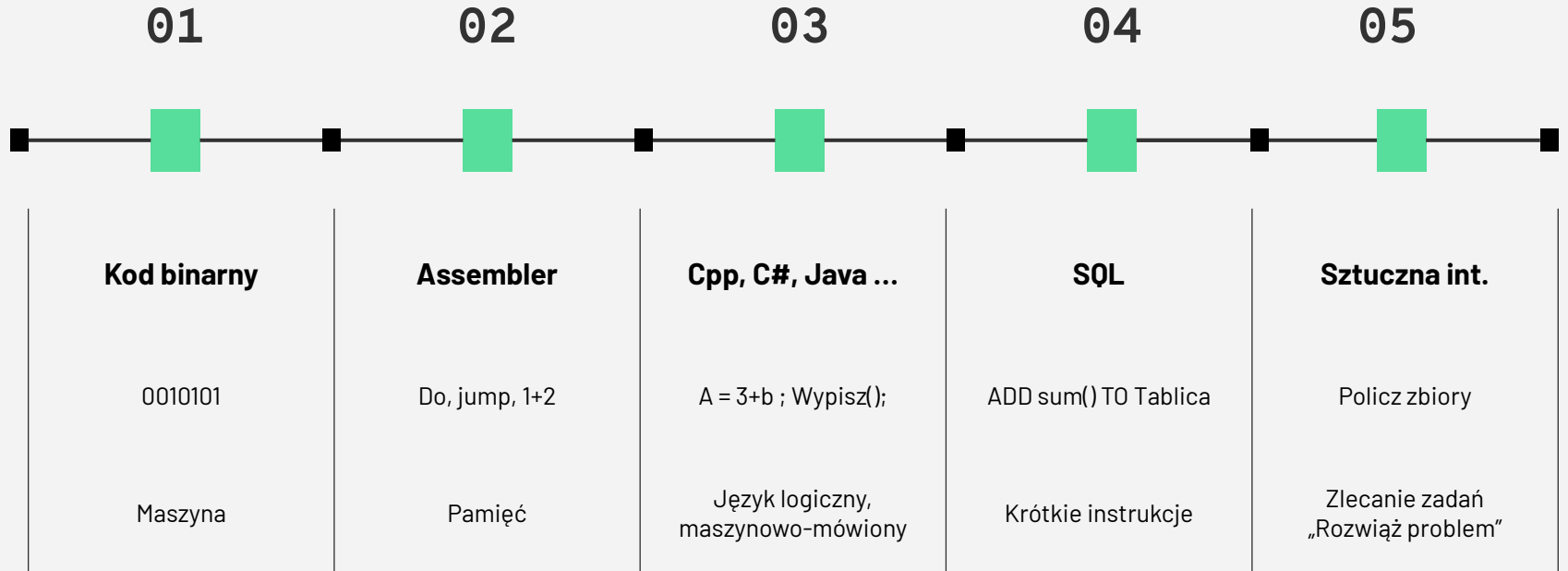
# Generacja języka

Określa poziom zaawansowania  
i samodzielności logiki języka programowania.

Im niższa generacja tym kod jest bardziej maszynowy.  
Im wyższa tym bardziej zbliżony do języka naturalnego  
(naszego mózgu).



# Generacje języków programowania



03

# Architektura

Wzorce projektowe

# Wzorce projektowe

Ang. Design Patterns

Gotowe wzory rozwiązania  
często napotykanym problemów  
programistycznych.

#Uniwersalne #Sprawdzone #Przetestowane #Wygodne

# Wzorce projektowe

## Korzyści:

// Pozwalają uniknąć typowych błędów i przyspieszają pracę

// Są uniwersalnym opisem rozwiązania problemu a nie jego implementacją więc mogą zostać użyte w każdym języku

// Istnieje wiele szkieletów projektów z zaimplementowanymi wzorcami w konkretnych językach więc możemy czerpać gotowców

z

## Podział wzorców projektowych

### Kreacyjne

Dotyczą sposobów tworzenia klas, metod, typów danych.

### Strukturalne

Dotyczą zależności powiązanych ze sobą obiektów

### Behawioralne

Dotyczą zachowania współpracujących ze sobą obiektów

# Podział wzorców projektowych

## Kreacyjne

Fabryka  
Budowniczy  
Prototyp  
Singleton

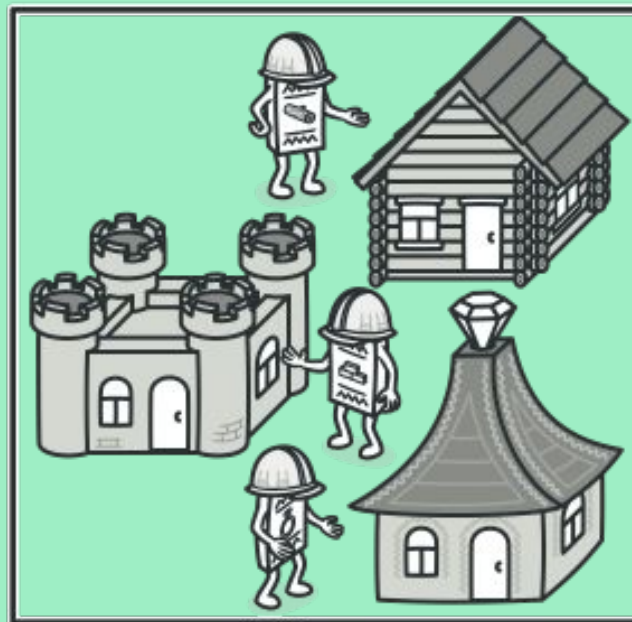
## Strukturalne

Dekorator  
Fasada  
Pyłek  
Most

## Behawioralne

Mediator  
Obserwator  
Iterator  
Pamiętka

# Problem A



ZbudujDom()

## Problem A2



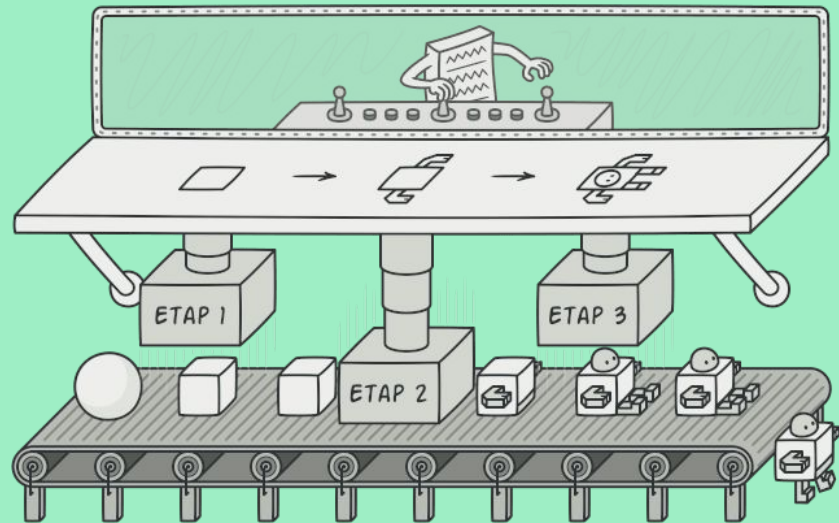
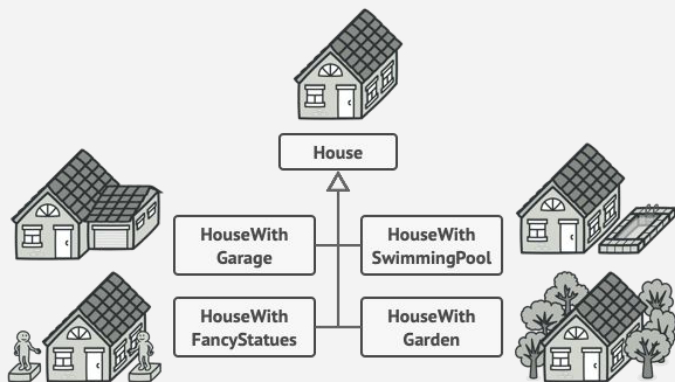
## Solution

Rozbicie zadania na etapy:  
PostawŚcianę(),  
WstawOkno(), PomalujŚcianę  
()..

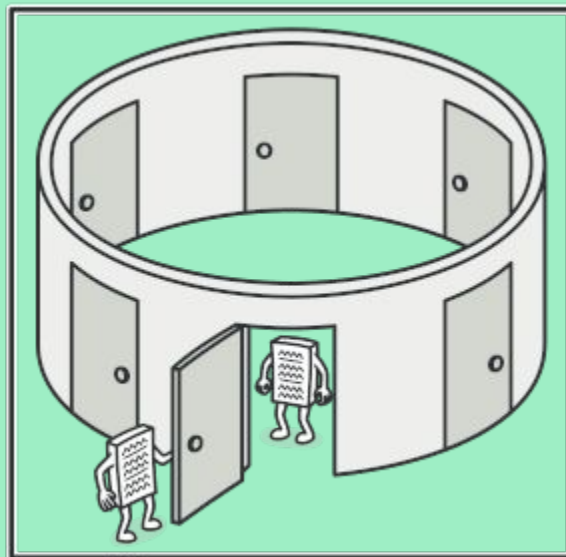


# Wzorzec Budowniczy

Daje możliwość tworzenia złożonych obiektów etapami, krok po kroku. Wzorzec ten pozwala produkować różne typy oraz reprezentacje obiektu używając tego samego kodu konstrukcyjnego.



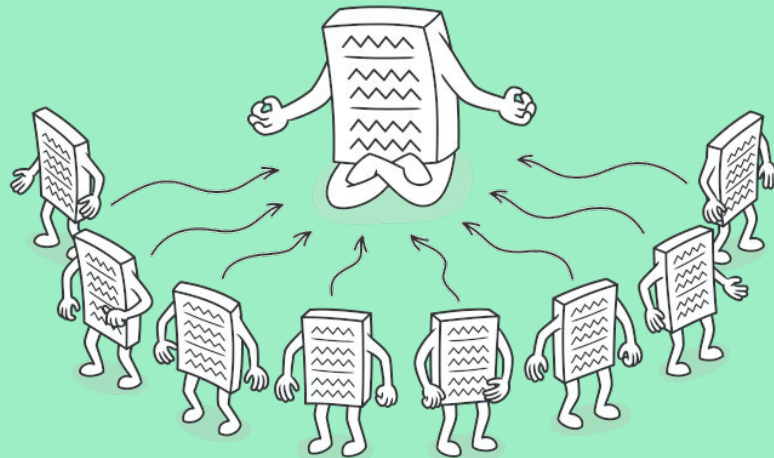
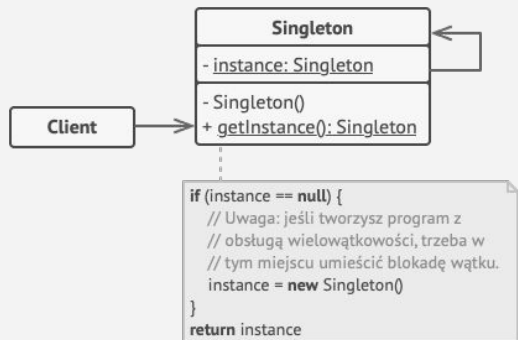
# Problem B



# Wzorzec Singleton

Pozwala zapewnić istnienie wyłącznie jednej instancji danej klasy. Ponadto daje globalny punkt dostępu do tejże instancji.

W skrócie - istnieje jeden kierownik, którego każdy prosi o dostęp do konkretnej rzeczy lub któremu każdy zadaje pytania.

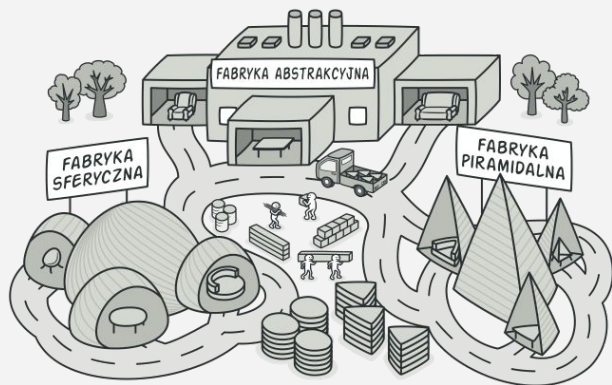


# Problem C



# Wzorzec Fabryka abstrakcyjna

Pozwala tworzyć rodziny spokrewnionych ze sobą obiektów bez określania ich konkretnych klas.



	Fotel	Sofa	Stolik Kawowy
Art Deco			
Wiktoriański			
Nowoczesny			

# Problem D

Front-end

Back-end



HTML

CSS

JavaScript

PHP

SQL

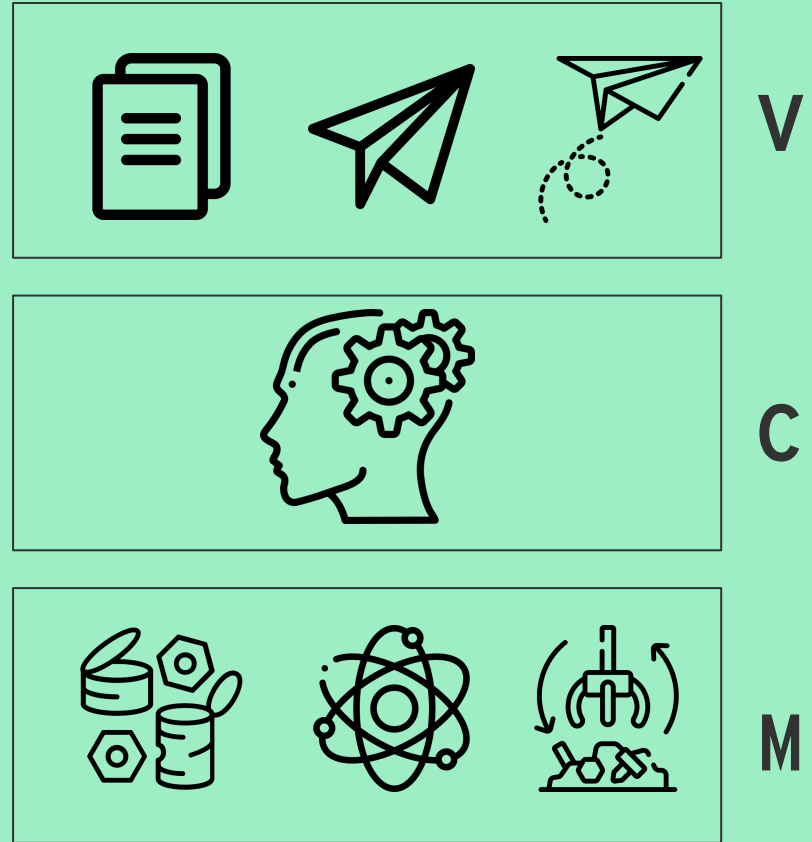
Java

# Wzorzec MVC

## Model View Controller

Służy do organizowania struktury aplikacji posiadających graficzne interfejsy użytkownika.

Jest połączeniem wzorców:  
Obserwator, Strategia i Kompozyt.



04

# Zadanie

Praca w grupach



# Zadania

01

Opisz wskazany język programowania. Jakiego jest rodzaju? Czym się charakteryzuje? Jakie ma zastosowania? Konstrukcja: zmiennej, warunku, pętli, funkcji, klasy.

02

Porównaj wskazane języki programowania.

03

Wskaż czym różnią się od siebie wskazane pojęcia.

# Zadanie

## Tematy dla grup

**A**

**B**

**C**

**01**

C++

Java

Python

**02**

C++ vs Python

Java vs C++

Python vs Java

**03**

Wskaźnik  
a Referencja

Stos  
Serta a

Współbieżność  
a Równoległość

# Thanks !

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik

Please keep this slide for attribution

Pytania? Zachęcam do kontaktu:

[pkucharczyk@technikumkreatywne.pl](mailto:pkucharczyk@technikumkreatywne.pl)

**kucharczyk.dev**

# Źródła

## Strony internetowe:

- [wikipedia.pl](https://wikipedia.pl)
- [jedlikowski.com](https://jedlikowski.com)
- [codenga.pl](https://codenga.pl)
- [refactoring.guru](https://refactoring.guru)

## Literatura:

- dr Przemysław Juszczuk  
“Języki programowania  
deklaratywnego”, Przemysław  
Juszczuk