

Wprowadzenie

Zaawansowane metody programowania

mgr inż. Krzysztof Rewak

15 lutego 2020

Wydział Nauk Technicznych i Ekonomicznych

Państwowa Wyższa Szkoła Zawodowa im. Witelona w Legnicy

Plan prezentacji

1. Ramowy plan semestru
2. Warunki zaliczenia kursu
3. Przegląd języków programowania
4. Podsumowanie

Ramowy plan semestru

Planowany rozkład jazdy

Wykład 1: Wprowadzenie; dobór technologii projektowej

Wykład 2: Zasady SOLID

Wykład 3: KISS, DRY, YAGNI i inne

Wykład 4: Czysty kod, część I

Wykład 5: Czysty kod, część II

Wykład 6: Czysty kod, część III

Wykład 7: Praktyczny polimorfizm

Planowany rozkład jazdy

Wykład 8: Programowanie asynchroniczne

Wykład 9: Wzorce projektowe: kreacyjne

Wykład 10: Wzorce projektowe: strukturalne

Wykład 11: Wzorce projektowe: operacyjne

Wykład 12: Antywzorce projektowe

Wykład 13: Programowanie ekstremalne

Wykład 14: Praca w zespole programistycznym

Wykład 15: Kolokwium zaliczeniowe

Warunki zaliczenia kursu



wykład - teoretyczna część kursu; przedstawienie wybranych zagadnień związanych z aplikacjami webowymi;

laboratorium - praktyczna część kursu; praca zespołowa nad projektowaniem, implementacją oraz wdrożeniem projektu programistycznego.

Warunki zaliczenia wykładu

Wykład kończy się egzaminem podsumowującym wiedzę przyswojoną w trakcie semestru. Egzamin - w zależności od liczby przystępujących osób - odbędzie się w formie pisemnej lub ustnej.

Student, który otrzyma z projektu ocenę niedostateczną nie może podchodzić do egzaminu.

Warunki zaliczenia wykładu

Ponadto na wykładach:

- będzie sprawdzana lista obecności na zasadzie białej listy;
- będzie mierzona (pozytywna i negatywna) aktywność studentów.

Zachęcam do uczęszczania na wykłady.

$$0.3W + 0.7L$$

Ocena niedostateczna z jednej formy rzutuje na ocenę niedostateczną za całość!

Przykładowo:

- $3.0 \text{ E} + 5.0 \text{ L} = 4.4 \Rightarrow \mathbf{4.5}$
- $4.5 \text{ E} + 4.0 \text{ L} = 4.15 \Rightarrow \mathbf{4.0}$
- $2.0 \text{ E} + 5.0 \text{ L} = 4.1 \Rightarrow \mathbf{2.0}$

Osoby, które otrzymały ocenę z projektu $p = 5.0$, zostaną zwolnione z egzaminu z przepisaną oceną.

Wysoka frekwencja oraz aktywność na wykładach mogą rzutować na obniżenie progu przepisanej oceny do $p \geq 4.5$ dla indywidualnych studentów.

W trakcie semestru organizowane będą dodatkowe zwolnienia z egzaminu w formie interaktywnych *kartków* z bieżącego materiału.

Student, który otrzyma najwięcej punktów z danej *kartkówki*, zostanie zwolniony z egzaminu z przepisaną oceną z projektu.

Bibliografia i ciekawe źródła



Robert C. Martin, *Czysty kod. Podręcznik dobrego programisty*, wyd. Helion



Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku*, wyd. Helion



PHP Design Patterns,
<https://designpatternsphp.readthedocs.io/en/latest/>

Pytania?

Dobór technologii projektowej

Zaawansowane metody programowania

mgr inż. Krzysztof Rewak

15 lutego 2020

Wydział Nauk Technicznych i Ekonomicznych

Państwowa Wyższa Szkoła Zawodowa im. Witelona w Legnicy

Wybór języka programowania może być jedną z najważniejszych projektowych decyzji.

Problematycznym podejściem może być sytuacja wprost ze znanego powiedzenia, w której mamy młotek i wszystko inne wygląda jak gwoździe.

Często się okaże, że nasz ulubiony język czy framework - mimo wszystko!
- nie będzie najlepszym z puli dostępnych rozwiązań.

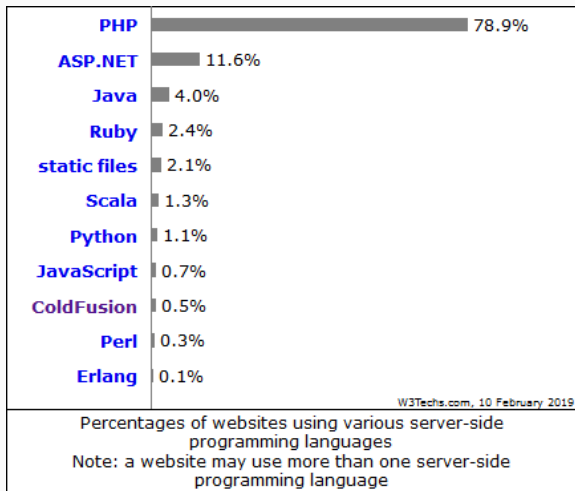
Przegląd języków programowania

Aplikacje webowe: backend

Webowy backend jest podstawą większości nowoczesnych aplikacji, zarówno klasycznych webowych, jak i desktopowych, mobilnych i innych.

SOAP, REST API, GraphQL czy też renderowanie HTML-owych widoków - każdy z tych sposobów komunikacji z backendem można zaimplementować w przynajmniej kilku różnych językach.

Aplikacje webowe: backend



Rysunek 1: [https:](https://w3techs.com/technologies/overview/programming_language/all)

[//w3techs.com/technologies/overview/programming_language/all](https://w3techs.com/technologies/overview/programming_language/all)

PHP jest naturalnym językiem programowania aplikacji webowych i powstał w tym celu w latach 90. ubiegłego wieku.

Cechy: obiektowy, dynamicznie i słabo typowany, interpretowany, wieloplatformowy, jeden proces na jedno zapytanie, mnóstwo frameworków.

Aplikacje webowe: backend w Javie

Java jest jednym z najczęściej używanych języków programowania na świecie, zarówno dla aplikacji mobilnych, desktopowych i webowych.

Cechy: obiektowa, statycznie typowana, kompilowana, wielowątkowa, wieloplatformowa.

Aplikacje webowe: backend w C#

C# jest językiem stworzonym przez Microsoft do realizacji wielu zadań, a dzięki frameworkowi ASP.MVC i Core można tworzyć rozległe aplikacje webowe.

Cechy: obiektowy, silnie typowany, kompilowany, wielowątkowy.

JavaScript wykształcił się jako webowa technologia do wspomagania frontendu, jednak w ostatnich latach wyewoluował w pełnoprawny język backendowy.

Cechy: wieloparadygmatowy, dynamicznie i słabo typowany, interpretowany, wieloplatformowy.

Aplikacje webowe: backend w Go

Go to odpowiedź Google na zapotrzebowanie rynku na szybki i wielowątkowy język programowania.

Cechy: statycznie i silnie typowany, kompilowalny, strukturalny, wielowątkowy.



Aplikacje webowe: frontend w JavaScriptcie

JavaScript obecnie jest praktycznie monopolistą na rynku frontendowych aplikacji webowych. Dzięki dziesiątkom frameworków, każdy znajdzie wygodnie rozwiązanie.

Cechy: wieloparadygmataowy, dynamicznie i słabo typowany, interpretowany, wieloplatformowy.

Aplikacje desktopowe



Aplikacje desktopowe: Java

Java to jeden z najpopularniejszych języków programowania użytkownych aplikacji desktopowych. Z produktami pracy z frameworkami takimi jak Swing czy JavaFX spotkał się prawie każdy użytkownik komputerów z Windowsem czy Linuksem.

Cechy: obiektowa, statycznie typowana, kompilowana, wielowątkowa, wieloplatformowa.

Aplikacje desktopowe: C#

C# z frameworkiem Microsoftu .NET jest podstawowym narzędziem do tworzenia windowsowych aplikacji.

Cechy: obiektowy, silnie typowany, kompilowany, wielowątkowy.

Aplikacje desktopowe: C++

C++ pozwoli na wykrzesanie z aplikacji największej wydajności, jednakże z pewnością będzie cięższy do opanowania. Popularnym frameworkiem do rozwiązań desktopowych wciąż jest Qt.

Cechy: imperatywny, obiektowy, statycznie i silnie typowany, kompilowany, wielowątkowy.

Aplikacje desktopowe: Python

Python ma multum frameworków pozwalających na tworzenie aplikacji desktopowych. Jest to ulubiony język do rozwiązań naukowych i okołomatematycznych ze względu na rozległe repozytoria bibliotek.

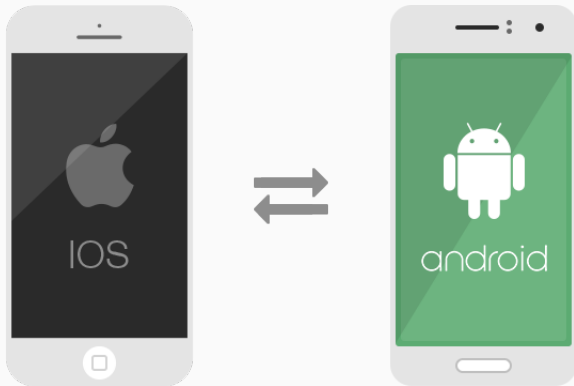
Cechy: interpretowany, dynamicznie i silnie typowany, obiektowy.

Aplikacje desktopowe: JavaScript

W **JavaScriptcie** można również tworzyć aplikacje desktopowe za pomocą specjalistycznych frameworków takich jak Electron.

Cechy: wieloparadygmata, dynamicznie i słabo typowany, interpretowany, wieloplatformowy.

Aplikacije mobilne



Aplikacje mobilne: Java

Java to obecnie królowa języków programowania mobilnego. Android to obecnie ponad 80% rynku smartfonów, a gigantyczna część wszystkich aplikacji napisana jest właśnie w Javie.

Cechy: obiektowa, statycznie typowana, kompilowana, wielowątkowa, wieloplatformowa.

Aplikacje mobilne: Kotlin

Kotlin to wschodząca gwiazda sponsorowana przez JetBrains i Google, a wykrozystująca JVM.

Cechy: statycznie typowany, kompilowany, wielowątkowy, wieloplatformowy.

Aplikacje mobilne: Swift

Swift wyparł Objective-C i stał się podstawowym językiem programowania systemów opartych iOS.

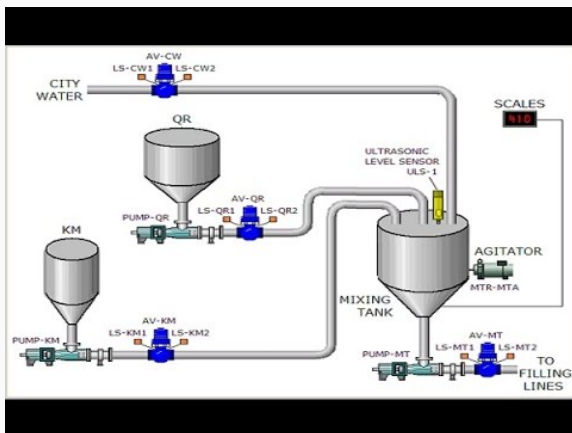
Cechy: wieloparadygmata, statycznie i silnie typowany, kompilowany.

Aplikacje mobilne: JavaScript

W **JavaScriptcie** można również tworzyć aplikacje mobilne za pomocą specjalistycznych frameworków takich jak NativeScript.

Cechy: wieloparadygmata, dynamicznie i słabo typowany, interpretowany, wieloplatformowy.

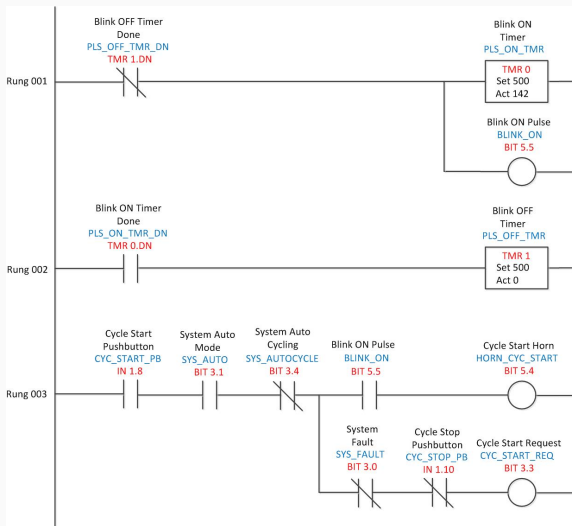
Aplikacje przemysłowe



Przemysł rządzi się swoimi prawami.

Aplikacje przemysłowe będą najczęściej programowane w językach niskiego poziomu lub w specjalistycznych językach typu LD.

Aplikacje przemysłowe



Podsumowanie

Pytania?

Kod prezentacji dostępny jest w repozytorium git pod adresem
<https://bitbucket.org/krewak/pwsz-zmp>



Wszystkie informacje dot. kursu dostępne są pod adresem
<http://pwsz.rewak.pl/kursy/10>

