



Politechnika
Wroclawska

Platformy programistyczne

Wprowadzenie

Dr inż. Radosław Idzikowski

Katedra Automatyki, Mechatroniki i Systemów Sterowania
Wydział Informatyki i Teleinformatyki

4 marca 2024



► Wykład

- Dr inż. Radosław Idzikowski,
p. 231A, bud. C-3,
radoslaw.idzikowski@pwr.edu.pl.

► Laboratorium:

- dr inż. Radosław Idzikowski,
p. 231A, bud. C-3,
radoslaw.idzikowski@pwr.edu.pl.
- dr inż. Aneta Górniak
aneta.gorniak@pwr.edu.pl.



O co chodzi?

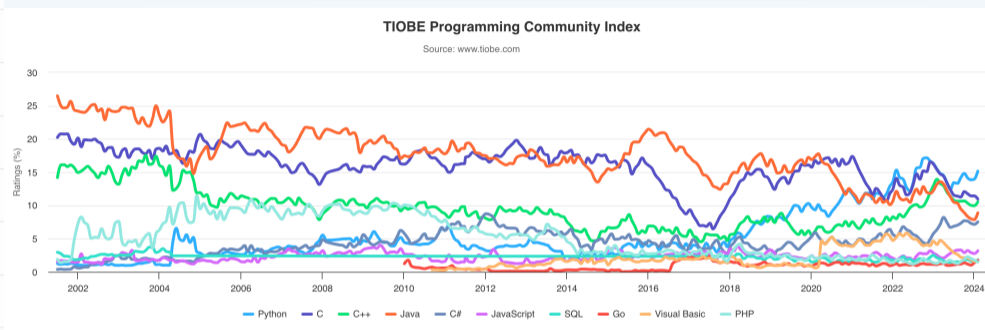


Zastosowania:

- ▶ aplikacje desktopowe,
- ▶ aplikacje webowe,
- ▶ rozwiązania klient/serwer,
- ▶ systemy wbudowane,
- ▶ aplikacje mobilne,
- ▶ aplikacje w chmurze,
- ▶ ...



TIOBE Index



Indeks Społeczności Programistycznej TIOBE to wskaźnik popularności języków programowania. Indeks jest aktualizowany raz w miesiącu. Oceny są oparte na liczbie wyspecjalizowanych inżynierów na całym świecie, kursów i dostawców zewnętrznych. Popularne strony internetowe, takie jak Google, Amazon, Wikipedia, Bing, oraz ponad 20 innych silników, są wykorzystywane do obliczania ocen.

Plan wykładu

1. Zajęcia wprowadzające.
2. Platforma .NET.
3. Typy danych i elementy programowania obiektowego.
4. Charakterystyka obiektowości, wyrażenia lambda, język LINQ.
5. Wykorzystanie platformy .NET w kontekście programowania równoległego oraz asynchroniczności.
6. Omówienie technologii ASP.NET.
7. Zaawansowane techniki programowania równoległego w platformie NET.
8. Kolokwium.

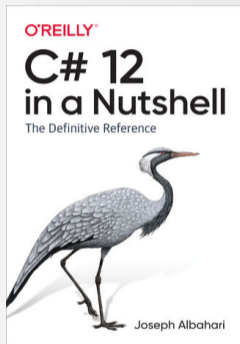
Zasady zaliczenia

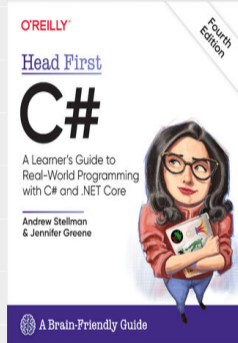
- ▶ $F1$ – ocena pozytywna z laboratorium,
- ▶ $F2$ – ocena pozytywna z kolokwium,
- ▶ P – ocena końcowa.

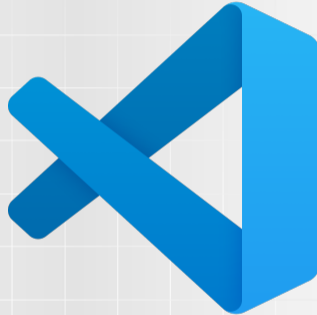
$$P = \frac{F1 + F2}{2}. \quad (1)$$

Konieczne jest uzyskanie oceny pozytywnej z każdej formy realizowanych w ramach przedmiotu (laboratorium oraz wykład).

- ▶ **Termin:** 04.06.2024,
- ▶ **Forma:** pytania otwarte,
- ▶ **Liczba pytań:** 15,
- ▶ **Czas trwania:** 35 min.







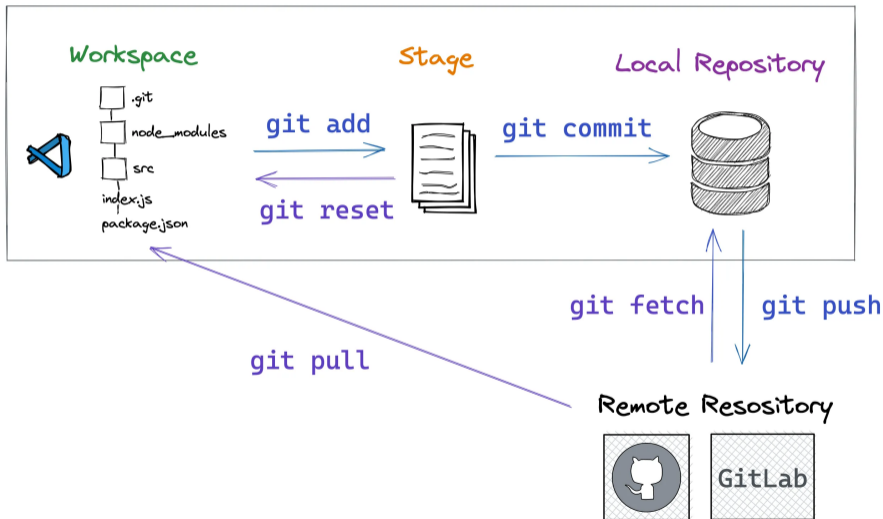
System kontroli wersji

- ▶ Oprogramowanie służące do śledzenia zmian w plikach (w szczególności w kodzie źródłowym).
- ▶ Pomaga zachować i przeglądać wprowadzane zmiany (a także porównywać wersje między sobą).
- ▶ Wspomaga łączenie zmian wprowadzanych przez różne osoby w zespole (a także w przez różne zespoły).
- ▶ Przykładowe systemy kontroli wersji: GIT, Subversion (SVN), Team Foundation Server (TFS).



Git

Local





Podstawowe komendy

- ▶ inicjalizacja nowego repo
`git init`
- ▶ klonowanie istniejącego
`git clone [url]`
- ▶ aktualny status
`git status`
- ▶ dodawanie zmian do stage
`git add <filename>`
- ▶ zakomitowanie zmian
`git commit -m "opis"`
- ▶ wrzucenie zmian na serwer
`git push origin <nazwa>`
- ▶ stworzenie brancha
`git branch <nazwa>`
- ▶ przełączanie brancha
`git checkout <nazwa>`