Praca zespołowa z Git

Czas: 2 × 45 min | praca w parach (osoba A i osoba B — role w scenariuszu) | Wykład 2.21

Cel: W parach poprowadzicie wspólne repozytorium na GitHub: gałęzie funkcji, pull requesty z przeglądem, kontrolowany konflikt scalania i ochronę gałęzi main.

Po zajęciach potrafisz:

- tworzy gałęzie funkcji i wypycha je do zdalnego repozytorium
- otwiera pull request, recenzuje i scala zmiany
- rozwiązuje konflikt scalania w Visual Studio
- konfiguruje ochronę gałęzi i szablon pull requesta

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu
- pendrive lub konto w repozytorium do zapisania projektu

Oprogramowanie

- **Visual Studio Community 2026 (obciążenie „Programowanie aplikacji klasycznych platformy .NET")** — darmowe dla uczniów i nauczycieli; w pracowni zainstalowane visualstudio.microsoft.com/pl/downloads
- **.NET 10 SDK (LTS)** — instalowany razem z Visual Studio; sprawdzenie: `dotnet --version` dotnet.microsoft.com/pl-pl/download/dotnet/10.0
- **Git for Windows** — lub Git wbudowany w Visual Studio git-scm.com/downloads/win
- **Konto GitHub (bezpłatne)** — lub szkolne konto GitHub Education github.com/signup

Bezpieczeństwo

- Zapisuj projekt w swoim folderze (np. `C:\Users\<login>\source\repos`), a na koniec zajęć skopiuj go na pendrive lub wyślij do repozytorium.
- Nie uruchamiaj programów i pakietów z nieznanych źródeł; pakiety NuGet instaluj tylko te, które wskazuje zadanie.
- Do logowania na GitHub używaj uwierzytelniania dwuskładnikowego; nie udostępniaj tokenów i haseł koledze.

Przebieg zadania

1 Repozytorium

Osoba A.

1. Wypycha aplikację ToDo (zadanie 6) na GitHub jako repozytorium prywatne; dodaje osobę B jako współpracownika.
2. Dodaje plik `.github/pull_request_template.md` (szablon w rozwiązaniu) i 2 zgłoszenia (Issues) z zadaniami.
3. Osoba B klonuje repozytorium w Visual Studio (Klonuj repozytorium).

2 Gałęzie i pull requesty

Praca równoległa (szczegóły: SCENARIUSZ_KONFLIKTU.md).

```
git switch -c funkcja/sortowanie
git add .
git commit -m "Sortowanie zadań po terminie (#2)"
git push -u origin funkcja/sortowanie
```

1. Każda osoba: gałąź `funkcja/…`, zmiana w innym miejscu kodu, commit z dobrym komunikatem, push.
2. Pull request z opisem i `Closes #numer`; druga osoba robi przegląd z co najmniej jednym komentarzem do linii kodu.
3. Po poprawce — merge i usunięcie gałęzi.

3 Konflikt

Celowo zmieniacie ten sam wiersz XAML.

1. Obie osoby zmieniają wiersz przycisku Dodaj na różne sposoby, każda na swojej gałęzi.
2. Osoba A scala pierwsza; osoba B scala main do swojej gałęzi i dostaje konflikt.
3. Rozwiązanie w Edytorze scalania VS, build, testy, commit, push, pull request.
4. Zrzut `git log --oneline --graph --all`.

4 Ochrona main

Ustawienia repozytorium.

1. Reguła: wymagany pull request i 1 akceptacja przed scaleniem do main.
2. Sprawdźcie, że bezpośredni push na main zostaje odrzucony — zapiszcie komunikat.

W sprawozdaniu umieść

- ☐ adres repozytorium i pull requestów
- ☐ zrzut komentarza z przeglądu
- ☐ zrzut edytora scalania i opis rozwiązania konfliktu
- ☐ graf historii
- ☐ komunikat odrzuconego pusha

Pytania kontrolne

1. Po co pracować na gałęziach?
2. Co sprawdzasz podczas przeglądu pull requesta?
3. Dlaczego konflikt powstał w kroku 3, a nie w kroku 2?
4. Co daje ochrona gałęzi main?

Kryteria oceny

Kryterium	Punkty
Repozytorium, szablon PR, zgłoszenia	3
Gałęzie i pull requesty z przeglądem	6
Rozwiązany konflikt	5
Ochrona main	3
Pytania	3
Razem	**20**

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).