Programowanie obiektowe 2023/2024

Laboratorium 9

Wersjonowanie kodu z pomocą systemu kontroli wersji

Wstęp

System kontroli wersji (Version Control System, VCS) to narzędzie, które umożliwia śledzenie zmian w kodzie źródłowym projektu, zarządzanie nimi i współpracę między członkami zespołu. Jest to niezwykle ważne narzędzie w pracy nad oprogramowaniem, ponieważ pozwala na skuteczne zarządzanie kodem, zachowanie historii zmian i współpracę nad projektem. Poniżej przedstawiam podstawowe kroki wersjonowania kodu przy użyciu systemu kontroli wersji, takiego jak Git:

1. Instalacja i konfiguracja systemu kontroli wersji:
   - Zainstaluj odpowiedni system kontroli wersji, na przykład Git, na swoim komputerze.
   - Skonfiguruj swoje dane użytkownika, takie jak imię i adres e-mail, aby system mógł identyfikować, kto dokonał danej zmiany.
2. Inicjowanie repozytorium:
   - Przejdź do katalogu, w którym chcesz przechowywać kod projektu.
   - Uruchom polecenie ***git init***, aby utworzyć nowe repozytorium Git.
3. Dodawanie plików do repozytorium:
   - Przejdź do katalogu projektu i utwórz lub skopiuj pliki źródłowe.
   - Użyj polecenia ***git add***, aby dodać pliki do indeksu (staging area), gotowe do zatwierdzenia.
4. Tworzenie commitów:
   - Użyj polecenia ***git commit***, aby utworzyć commit (zatwierdzenie) z dodanymi plikami. Dodaj krótki opis zmiany w komunikacie commita.
5. Śledzenie historii i porównywanie wersji:
   - Użyj polecenia ***git log***, aby przeglądać historię commitów i zobaczyć, kto i kiedy wprowadzał zmiany.
   - Możesz również użyć ***git diff***, aby porównać różnice między różnymi wersjami plików.
6. Praca z gałęziami (branches):
   - Gałęzie pozwalają na równoczesną pracę nad różnymi funkcjonalnościami projektu. Użyj ***git branch***, aby stworzyć nową gałąź, a następnie przełącz się na nią przy pomocy ***git checkout***.
7. Współpraca zespołowa:
   - Możesz współpracować z innymi członkami zespołu, wysyłając i pobierając zmiany z repozytorium zdalnego, takiego jak repozytorium na platformie GitHub, GitLab lub Bitbucket.
8. Rozwiązywanie konfliktów:
   - Jeśli dwie lub więcej zmian koliduje ze sobą, musisz rozwiązać konflikty ręcznie, a następnie utworzyć nowy commit.
9. Aktualizacja repozytorium:

- Regularnie aktualizuj swoje lokalne repozytorium przy pomocy ***git pull***, aby pobrać najnowsze zmiany ze zdalnego repozytorium.

10. Udostępnianie swoich zmian:
    - Po zakończeniu pracy nad daną funkcjonalnością lub poprawką, użyj ***git push***, aby wysłać swoje zmiany do repozytorium zdalnego.

Systemy kontroli wersji, takie jak Git, pozwalają na efektywne zarządzanie kodem źródłowym, monitorowanie zmian i współpracę w zespołach programistycznych. To ważne narzędzie dla każdego projektu programistycznego.

Zadania

Zadanie 1. Utwórz konto na platformie https://github.com

Zadanie 2. Utwórz repozytorium zawierające Twój numer albumu, nazwę przedmiotu, grupę oraz semestr np.: w11111_ProgramowanieObiektowe_GL01_3IID-2020

Zadanie 3. Zamieść link do repozytorium w zadaniu na platformie moodle. Jeśli repozytorium jest prywatne dodaj dostęp (settings/collaborators) prowadzącemu wykorzystując jego maila.

Zadanie 4. Przy użyciu polecenia git clone sklonuj repozytorium zdalne. Utwórz w repozytorium plik .gitignore. Do tego pliku dodaj konfigurację z poniższego linku:

https://github.com/github/gitignore/blob/main/VisualStudio.gitignore

Zadanie 5. Dodaj plik zawierający twoje dane.

Zadanie 6. Utwórz nową gałąź, zmodyfikuj plik. Na głównej gałęzi wprowadź zmiany które będą się wykluczały ze zmianami z nowej gałęzi. Scal zmiany z obu gałęzi rozwiązując konflikty.