Zadania dotyczące Gita i GitHuba
Zadania do zrealizowania w dniu 25.11.2021 15-20

Zadanie 1. Zainstaluj Gita na swoim komputerze. Sprawdź, czy Git został poprawnie zainstalowany poprzez sprawdzenie jego wersji (należy użyć odpowiedniej komendy w konsoli Windowsa).

Zadanie 2. Stwórz konto w serwisie GitHub, a następnie zaloguj się na nie w przeglądarce. W odpowiednich opcjach na GitHubie wygeneruj token dostępowy (będzie on potrzebny w momencie łączenia repozytorium lokalnego z repozytorium zdalnym w serwisie GitHub).

Zadanie 3. Zainstaluj jeden z darmowych programów z graficznym interfejsem użytkownika (GUI) do zarządzania gitem (np. GitKraken lub GitHub Desktop). Skonfiguruj podstawowe opcje (nazwa użytkownika, e-mail), połącz go z kontem w serwisie GitHub oraz zapoznaj się z interfejsem programu.

W razie korzystania z programu GitKraken, w celu integracji z GitHubem, pomocny może okazać się następujący link: `https://www.gitkraken.com/integrations/github`

Zadania do wykonania w aplikacji z GUI do zarządzania gitem

Zadanie 4. Stwórz nowe repozytorium lokalne w programie do zarządzania gitem (np. GitKraken) oraz dodaj do repozytorium kilka plików tekstowych. Następnie wykonaj commita i zobacz zmiany. Zmodyfikuj pliki tworząc kilka kolejnych commitów. Obserwuj zmiany.

Zadanie 5. Stwórz nowe repozytorium w serwisie GitHub i połącz je z lokalnym repozytorium z poprzedniego zadania. Zobacz, czy w zdalnym repozytorium na GitHubie zostały odnotowane wszystkie zmiany (przeglądnij historię commitów).

Zadanie 6. Kontynuując pracę w zdalnym repozytorium z poprzednich dwóch zadań, zmodyfikuj pliki, dodaj kilka commitów, a następnie prześlij zmiany do zdalnego repozytorium. Sprawdź zdalne repozytorium na GitHubie.

(dla zaawansowanych - nieobowiązkowe). Następnie w lokalnym repozytorium utwórz nowy branch o nazwie `logowanie`, dodaj do niego kilka plików tekstowych. Pracując na branchu `logowanie` utwórz kilka commitów.

Zadanie 7. (dla zaawansowanych - nieobowiązkowe). W tym zadaniu dalej znajdujemy się w lokalnym repozytorium z poprzednich zadań. Połączmy teraz branch (gałąź) `logowanie` z branchem master. Dodajmy commita i całość zmian wypchnijmy do zdalnego repozytorium.

Zadanie 8. Sklonuj repozytorium z projektem Delphi z adresu url: `https://github.com/rafalbrociek/DelphiApp.git`. Uruchom skopiowany program w środowisku programistycznym.

Zadania do wykonania z poziomu konsoli Gita

Zadanie 9. Z poziomu konsoli Gita skonfiguruj swoją nazwę użytkownika i email.

Zadanie 10. Stwórz na dysku katalog, a w nim utwórz lokalne repozytorium gita. Czy w katalogu zostały utworzone jakieś pliki? Wyświetl w konsoli listę plików, sprawdź, status repozytorium.

Zadanie 11. W katalogu repozytorium (z poprzedniego zadania) dodaj cztery pliki tekstowe: `plikA.txt, plikB.txt, plikC.txt, plikD.txt`. Jak tworzyć pliki z poziomu konsoli? Następnie dodaj pliki `plikA.txt, plikB.txt, plikC.txt` do staging area oraz wykonaj swojego pierwszego commita odpowiednio go komentując. W kolejnym kroku usuń plik `plikD.txt` oraz zmodyfikuj zawartość pliku `plikA.txt`. Wykonaj kolejny commit oraz wyświetl listę wszystkich commitów.

Zadanie 12. Dobrym pomysłem jest poinformowanie Gita, które pliki powinien śledzić, a których nie. Często pliki automatycznie generowane przez środowisko programistyczne nie powinny być uwzględniane przez Gita. Dodaj do repozytorium trzy pliki excela `koszty.xlsx, budzet.xlsx, wycena.xlsx`, dwa pliki worda `oswiadczenie.docx, raport.docx` oraz folder `important`, w którym dodajmy kilka plików tekstowych. Następnie skonfiguruj plik `.gitignore` w ten sposób, aby Git nie brał pod uwagę plików o rozszerzeniu `.xlsx`, pliku `odwiadczenie.docx` oraz katalogu `important`. Sprawdź status repozytorium.

Zadanie 13. (dla zaawansowanych - nieobowiązkowe). Pracujesz w repozytorium nad projektem na głównej gałęzi (branchu) `master`. Nagle masz za zadanie dodać nową funkcjonalność do projektu. W tym celu tworzysz nowy branch o nazwie `nowa-funkcjonalnosc`. Na nowej gałęzi tworzysz dwa foldery, dodajesz do nich pliki i je modyfikujesz. Stwierdzasz, że praca nad nową funkcjonalnością została skończona, po czym commitujesz zmiany. Połącz branch `nowa-funkcjonalnosc` z branchem `master`, a następnie skasuj branch `nowa-funkcjonalnosc`. Sprawdź status repozytorium i historię commitów.

Zadanie 14. (dla zaawansowanych - nieobowiązkowe). Pracujesz w repozytorium nad projektem na głównej gałęzi (branchu) `master`. Utwórz nowy branch `szyfrowanie` na nową funkcjonalność dotyczącą szyfrowania. Dodaj na tej gałęzi kilka plików, po całości prac wykonaj commita. Przełącz się na gałąź `master`. Sprawdź ile jest branchów w repozytorium i na jakim aktualnie branchu jesteś. Teraz utwórz nowy branch `logowanie` i i dodaj na nim kilka plików i katalogów. Po skończonej pracy zrób commita. Następnie połącz wszystkie trzy gałęzie do branchu `master`.

Zadanie 15. (dla zaawansowanych - nieobowiązkowe). Pracujesz w repozytorium nad projektem na głównej gałęzi (branchu) `master`. Utwórz dwa nowe pliki tekstowe `A.txt, B.txt`. Zmodyfikuj te pliki, zrób commita. Następnie stwórz kolejny plik tekstowy `C.txt` i również go zmodyfikuj, po czym zrób commita. Sprawdź historię commitów. Stwierdzasz, że plik `C.txt`, jak również commit po jego modyfikacji były niepotrzebne. Powróć ze stanem repozytorium do pierwszego z commitów. **Wskazówka. Opcja** `git reset`. Spróbuj wykonać to zadanie z poziomu programu graficznego.

Zadanie 16. Stwórz na GitHubie repozytorium (zwróć uwagę, czy repozytorium ma być prywatne, czy publiczne). Następnie na komputerze w określonej przez siebie lokalizacji sklonuj repozytorium zdalne z GitHuba. Pamiętaj o odpowiedniej nazwie użytkownika oraz tokenie dostępowym.

Zadanie 17. Kontynuując wątek z zadania poprzedniego - dodaj w lokalnym repozytorium kilka plików i folderów (pliki zmodyfikuj dodając do nich treść). Następnie zrób kilka commitów. Po zrobieniu commitów (i sprawdzeniu ich historii) wypchnij wszystkie zmiany do zewnętrznego (zdalnego) repozytorium na GitHubie. Sprawdź w serwisie GitHub, czy operacja powiodła się i widać tam wszystkie zmiany.

W ramach potwierdzenia realizacji zadań proszę o umieszczenie wpisu z linkiem do stworzonego przez Państwa dowolnego repozytorium zdalnego w serwisie GitHub w zespole „Programista w języku Delphi" na kanale „Git" (zakładka Wiki) w dniu 25.11.2021

Pytania pomocnicze

— Czym jest system kontroli wersji?
— Jakie wyróżniamy trzy główne typy systemu kontroli wersji?
— Czym różni się scentralizowany system kontroli wersji od rozproszonego?
— Czym jest GitHub?
— Co oznaczają pojęcia: working directory (przestrzeń robocza), staging area (poczekalnia), repository (repozytorium), commit oraz branch (gałąź)?
— Do czego służy plik `.gitignore`?