

Podstawowy warsztat informatyka — lista 5

Zadanie 1. (1 punkt) Stwórz katalog `testy`. Utwórz w nim puste repozytorium. Utwórz w nim plik `README.md` o treści „Zadanie 1 z listy 5 z PWI.”. Skomituuj¹ ten plik (do gałęzi `master`). Dopisz do pliku `README.md` swoje imię i nazwisko i skomituuj jeszcze raz. Zastanów się, jaka jest najciekawsza rzecz, którą dotychczas poznałeś/poznałaś na naszych studiach. Dopisz tę informację (w nowej linii) do pliku `README.md` i ponownie skomituuj zmiany. W kolejnej linii wpisz nazwę czegoś, co regularnie czytasz/słuchasz/oglądasz (czasopismo, podcast itp.). Znow skomituuj zmiany.

Wpisz `git log`, aby zobaczy utworzony komity. Wykorzystaj polecenie `git checkout` do przełączenia się między komitami, używając ich kodów (nie trzeba wpisywać całego haszu komita, tylko pierwszych parę znaków, np. `git checkout 41d5`, ale dobrą praktyką jest podawanie przynajmniej 7 znaków). Na koniec przełącz się (tym samym poleceniem) do gałęzi `master`.

Stwórz drugi plik, o nazwie `skrypt.py` i o treści:

```
liczba = 5
silnia = 1
while liczba > 0:
    silnia *= n
    liczba -= 1
print(silnia)
```

Skomituuj ten plik (do gałęzi `master`). Następnie popraw oczywisty błąd w skrypcie i stwórz kolejny komit, w którym skrypt `skrypt.py` poprawnie liczy 5!

Zadanie 2. (2 punkty) *Sklonuj* na swój dysk repozytorium:

<https://github.com/alewmoose/2048-in-terminal>

Skompiluj grę poleceniem `make`². Uruchom grę i pograj trochę – zdobądź ponad 600 punktów. Wyłącz grę i przełącz się na pliki źródłowe. Znajdź plik, w którym ustawione są etykiety kafelków (2, 4, ...). Zamień te etykiety na składające się wyłącznie z liter, następnie ponownie skompiluj grę. Czy kompilacja tym razem była szybsza? Pokaż prowadzącemu efekt swoich działań - planszę ze zmodyfikowanymi kafelkami i wynikiem ponad 600 punktów.

W katalogu gry użyj polecenia `git log`. Co się stało? Następnie wypróbuj (ze zrozumieniem) polecenia `git log --pretty=oneline` i `git blame README.md`.

Zadanie 3. (2 punkty) Wróć do katalogu `2048-in-terminal` z poprzedniego zadania. Wykonaj polecenie `git stash`, żeby odrzucić dotychczasowe zmiany. Następnie wykonaj następujące polecenia:

```
git rm LICENSE
git commit -m "license removed"
less LICENSE
rm screen.gif
git commit -m "screen.gif removed"
less screen.gif
```

Dowiedz się (ale nie zgaduj, tylko faktycznie się dowiedz), dlaczego tak się stało. Zastanów się, czy teraz polecenie `git add screen.gif`, po którym nastąpi odpowiedni komit, doda ten plik do repozytorium, czy wręcz przeciwnie – skasuje go.

Teraz trzeba usunąć te niepotrzebne komity. Wykonaj polecenia:

```
git log
git reset --hard b
git reset --hard bf
git reset --hard bf2
git reset --hard bf22
git log
```

¹Anglicyzmy są na tym przedmiocie niestety nieuniknione. Jeśli ktoś powie, że „zatwierdza zmiany w składnicy”, to nikt go nie zrozumie.

²Na serwerze `pwi` `make` nie jest zainstalowany

Jaki był efekt i dlaczego?

Czas zrobić coś głupiego. Wyдай polecenie:

```
git revert -m 1 4378..HEAD
```

Zobacz poleceniem `git log`, że stało się coś dziwnego. Użyj polecenia `git reset`, aby posprzątać.

Zadanie 4. (1 punkt) Jeśli nie masz konta w serwisie github, to takie konto załóż. Umieść na serwerze github swój klucz publiczny, a by nie musieć podawać hasła przy każdym połączeniu. Następnie wejdź pod adres tego zadania <https://classroom.github.com/a/HqUxNNII>, zaloguj się swoim kontem w serwisie github i wybierz zespół „wszyscy”. Dzięki temu dołączysz do grupy stworzonej na potrzeby tego przedmiotu i zyskasz dostęp do repozytorium tego zadania. W repozytorium tym znajduje się plik README.md. Znajdź go, przeczytaj polecenia i następnie je wykonaj.