za<pro/>gramowani.com

</> webmaster

</> kontakt

E-mail: hello@filiposinski.com

Discord: https://discord.gg/eyhAtuxJcv

GitHub: github.com/zaprogramowaniFO

</> Zajęcia **nr 5**

2023/11/21

</ /> GITHUB

GIT, czyli **rozproszony system kontroli wersji**

https://git-scm.com

Stworzony przez **Linusa Torvaldsa**, twórcę **Linuxa**.
Pierwsza wersja Gita pojawiła się w kwietniu 2005 roku.

Git to typ oprogramowania, który służy przede wszystkim do śledzenia zmian w naszym kodzie.

GIT, czyli **rozproszony system kontroli wersji**

Jeżeli:

- Wczoraj działało, a dzisiaj już nie,
- Chcemy wrócić do poprzedniej wersji kodu,
- Chcemy wprowadzić parę zmian, ale jednocześnie móc mieć obecną wersję programu,
- Chcemy pracować nad projektem w grupie, to...

... pomoże nam w tym Git ;)

GitHub - czyli Twoje programistyczne portfolio

Jeżeli:

- Wczoraj działało, a dzisiaj już nie,
- Chcemy wrócić do poprzedniej wersji kodu,
- Chcemy wprowadzić parę zmian, ale jednocześnie móc mieć obecną wersję programu,
- Chcemy pracować nad projektem w grupie, to...

... pomoże nam w tym Git ;)

GIT, czyli **rozproszony system kontroli wersji**

https://github.com

Serwis działa od kwietnia 2008 roku.

Jest to serwis, który pozwala na przechowywanie repozytoriów na swojej przestrzeni serwerowej.

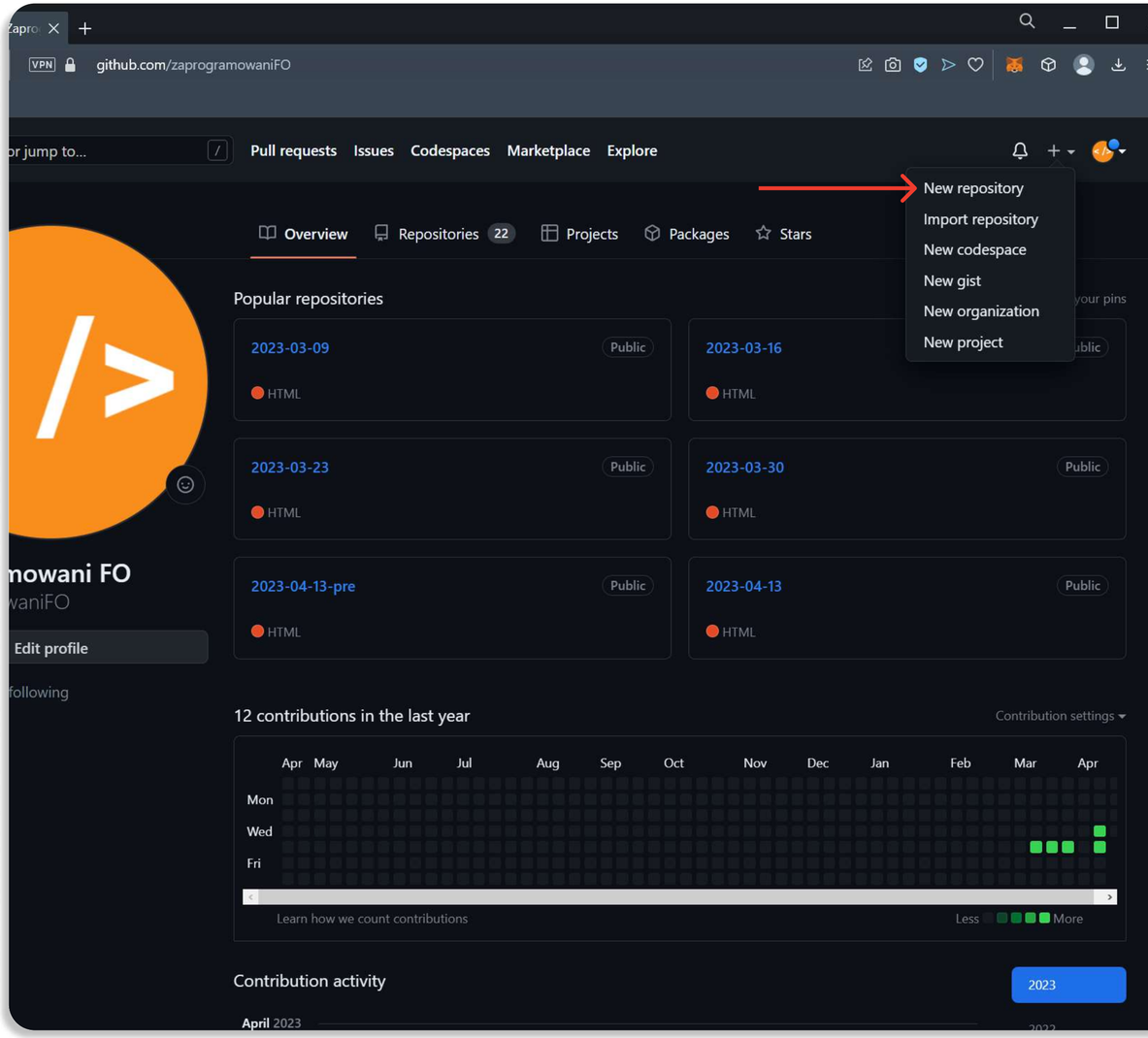
Dzięki niemu wszystkie swoje projekty możesz opublikować oraz na bieżąco wprowadzać zmiany.

</> GIT != Github

Tworzymy repozytorium na GITHUBIE

Tworzymy repozytorium

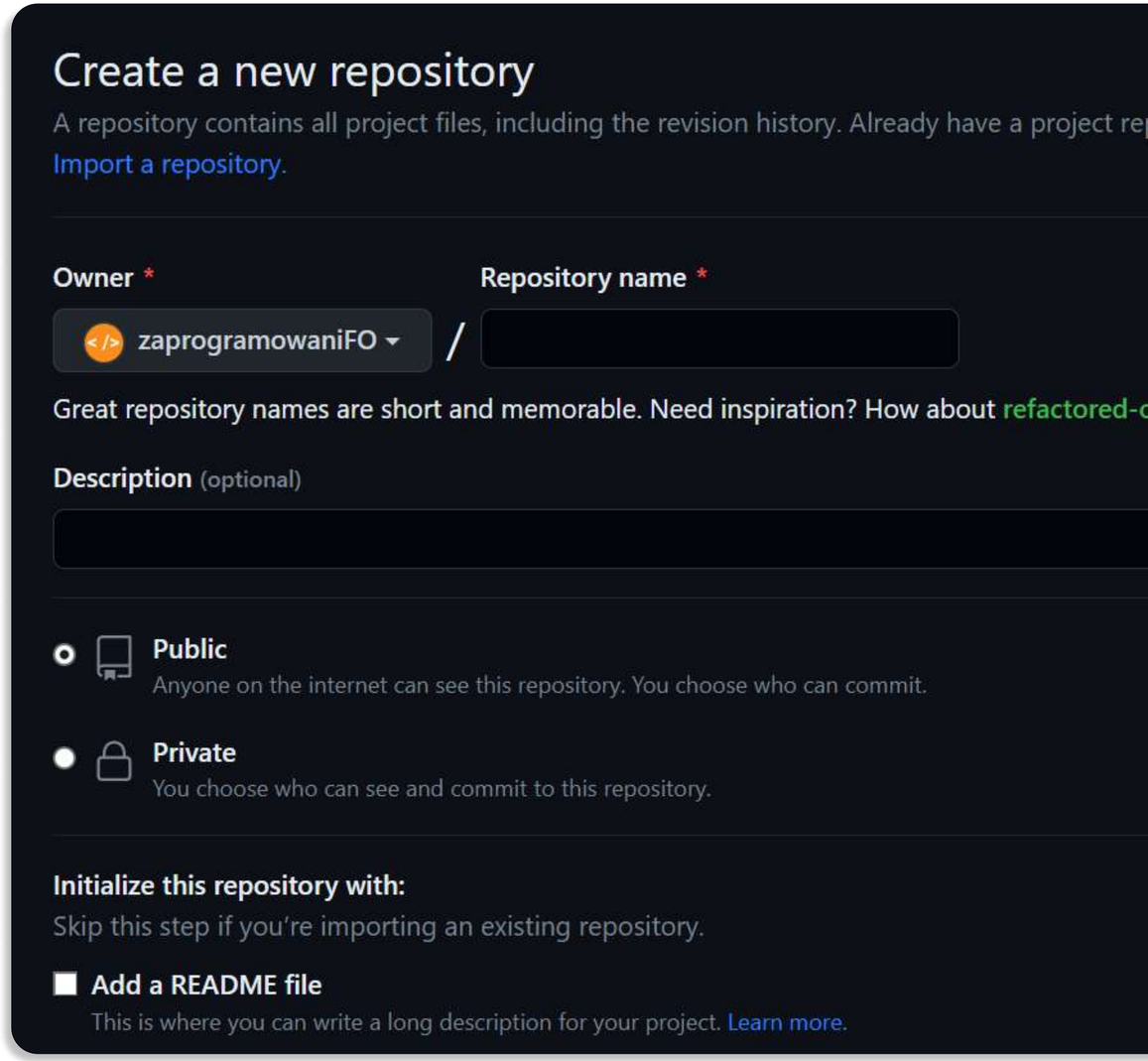
Będąc na naszym profilu na **Githubie**, klikamy w **plus** obok naszego logo w prawym górnym narożniku, a następnie z menu kontekstowego wybieramy **new repository.**



Tworzymy repozytorium

Wpisujemy **nazwę naszego repozytorium (repository name)**, możemy również dodać od razu opcjonalny opis.

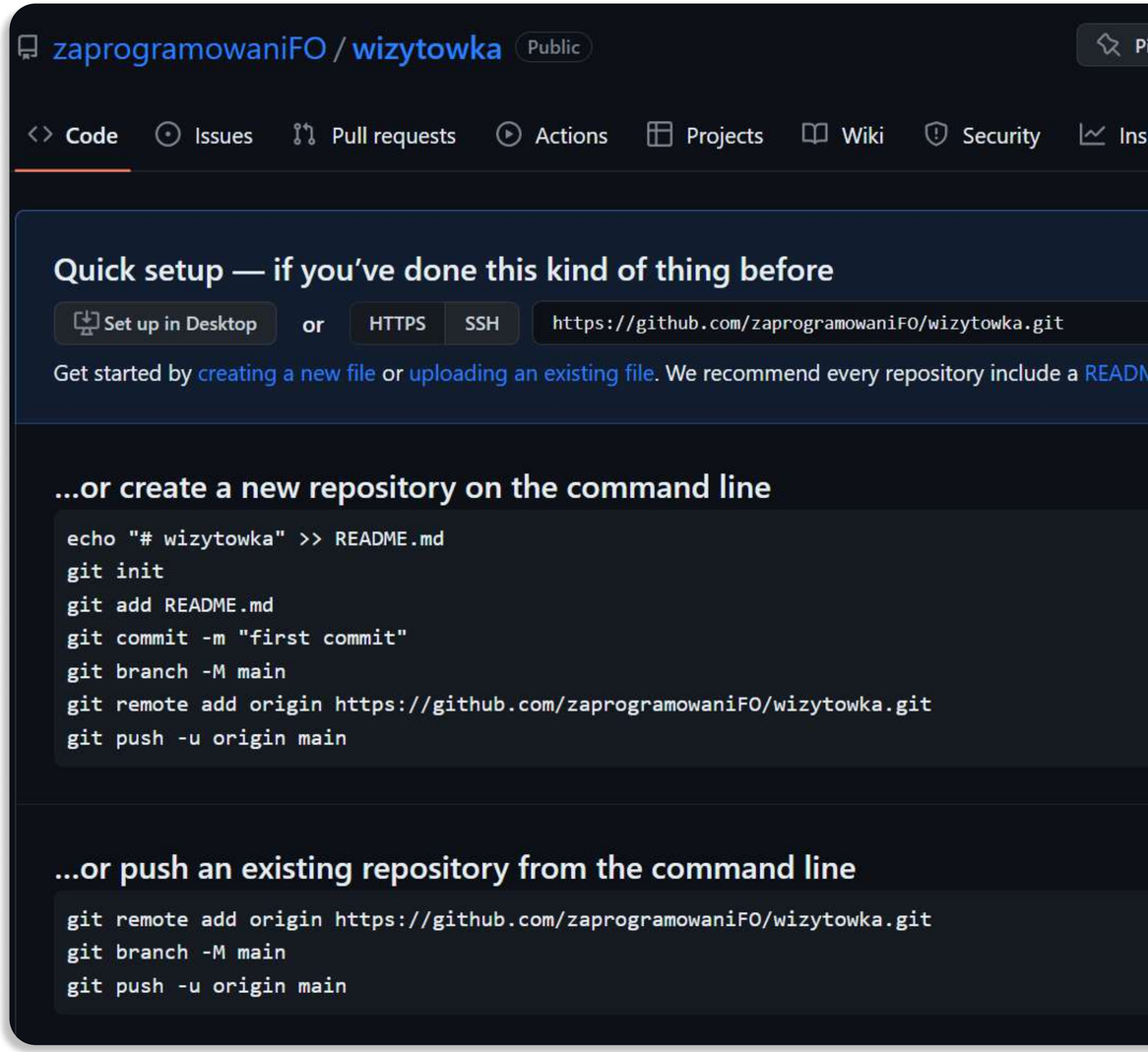
Jeżeli chcemy skorzystać z **hostingu** oferowanego przez **github**, nasze repozytorium musi być publiczne.



Tworzymy repozytorium

Następnie powinniśmy podpiąć repozytorium utworzone na naszym komputerze, do stworzoneg repozytorium na platformie github, aby w prosty sposób synchronizować zmiany w projekcie.

Możemy również w prosty sposób umieszczać bezpośrednio pliki w repozytorium na stronie.
Klikamy upload an existing file.



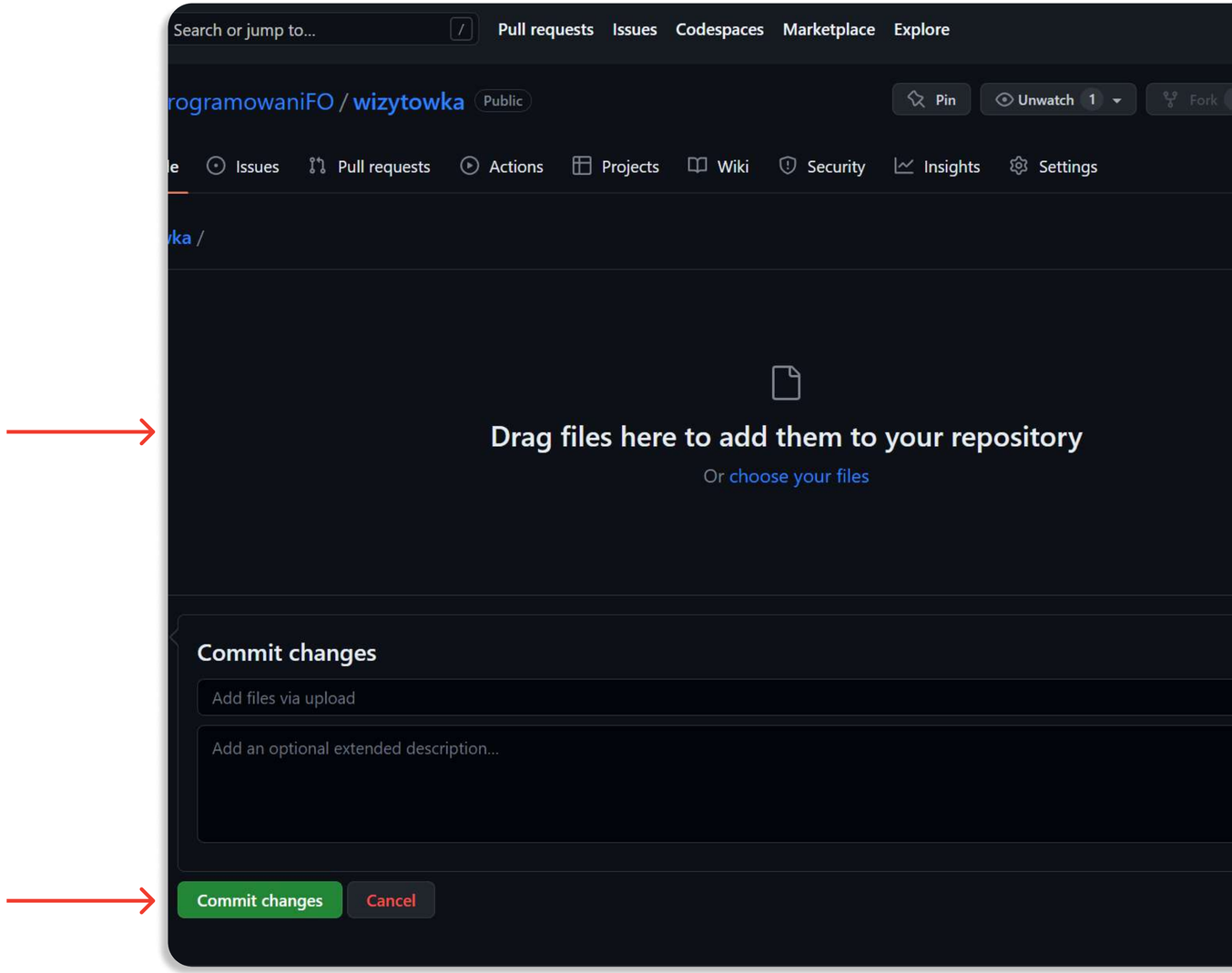
Tworzymy repozytorium

W tym miejscu możemy bezpośrednio umieszczać pliki poprzez przeciąganie.

Możemy też napisać komentarz do naszego **commita**.

Po umieszczeniu plików należy kliknąć **commit changes**.

W ten sposób zrobiliśmy naszego pierwszego **commita** w projekcie! :)

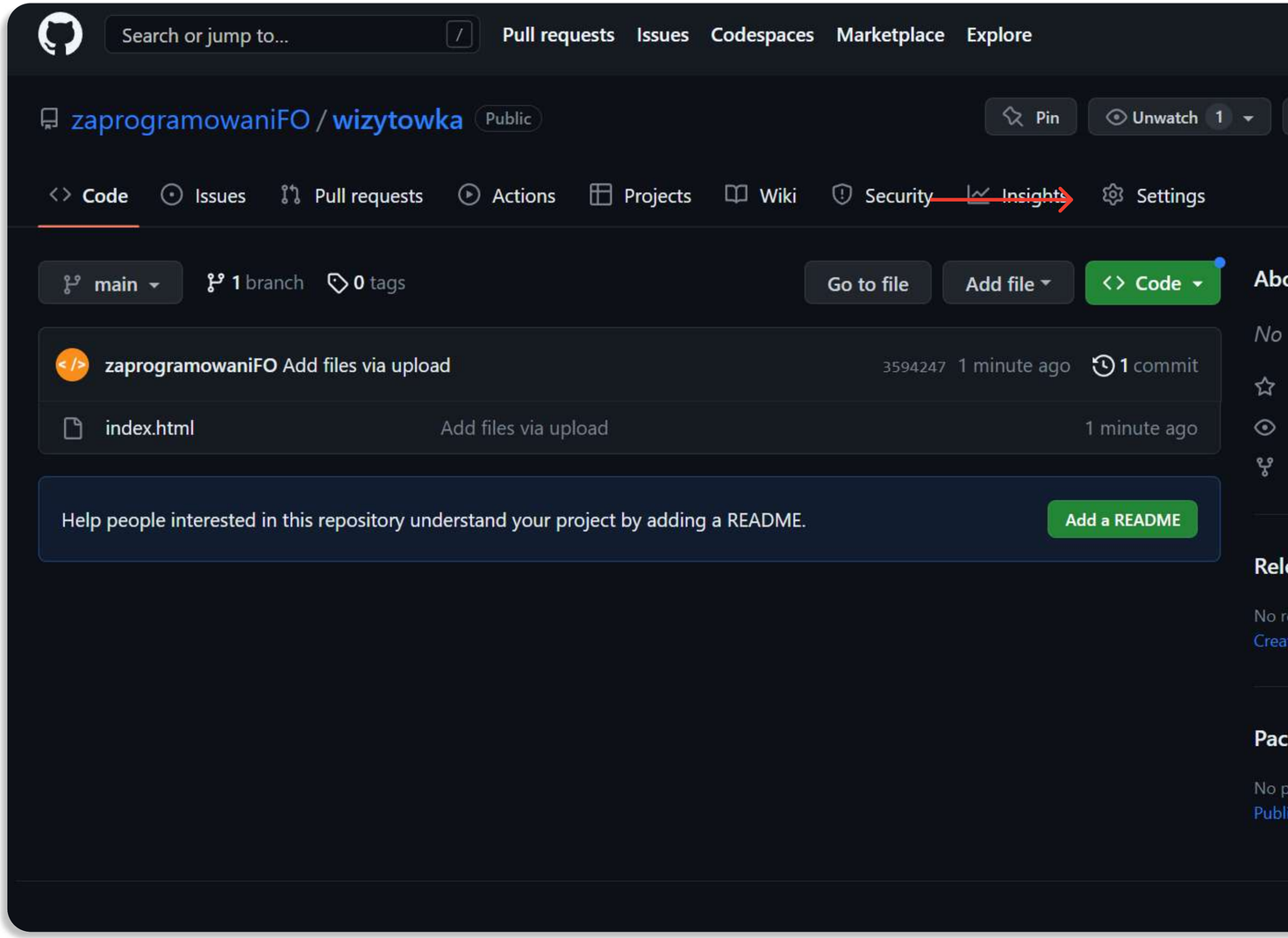


Publikujemy stronę na GITHUBIE

Publikujemy stronę

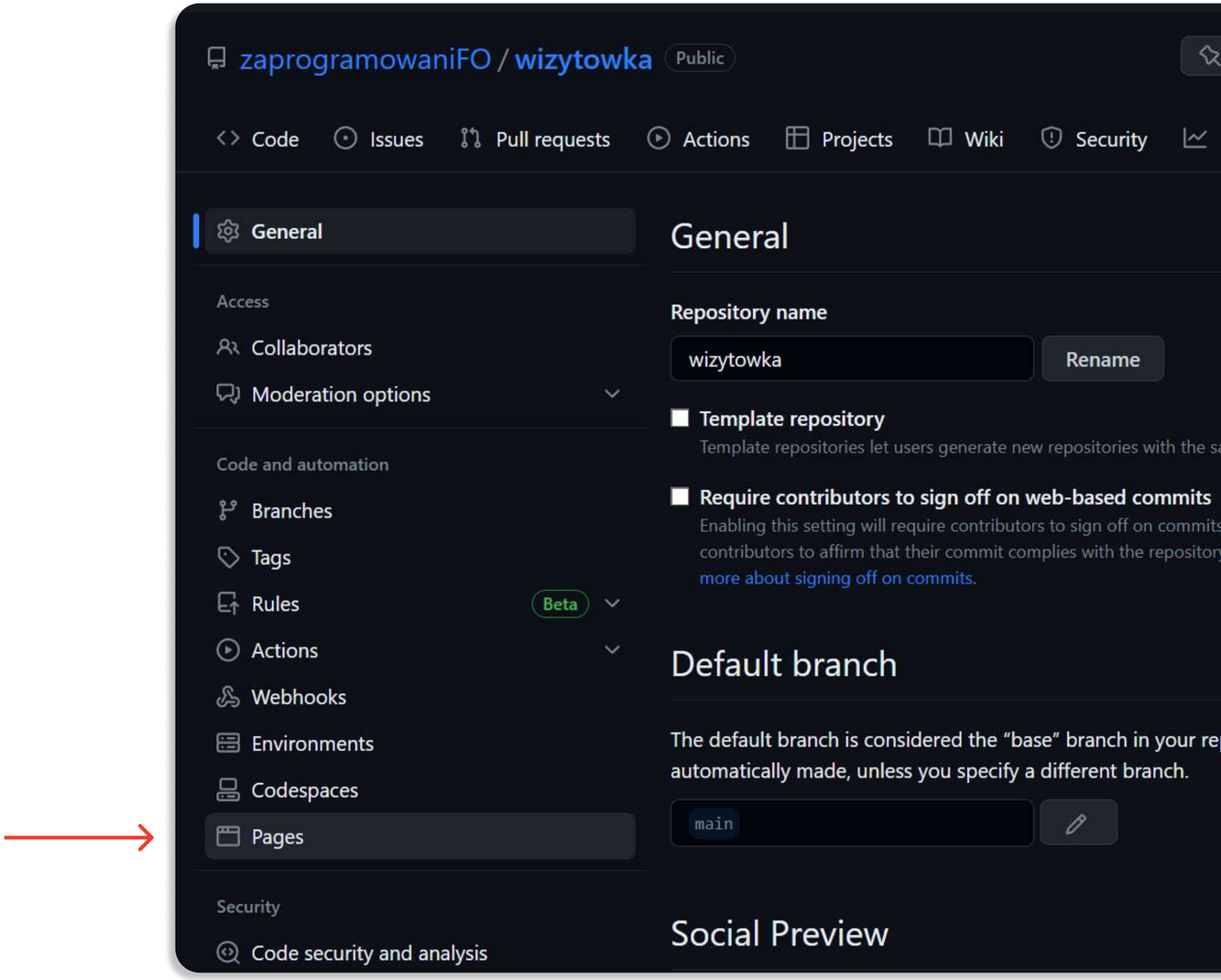
Od publikacji strony dzieli nas jeszcze kilka kroków!

W gotowym repozytorium, które chcemy udostępnić jako stronę, klikamy w **ustawienia (settings)**.



Publikujemy stronę

Następnie przechodzimy do sekcji **Pages**.

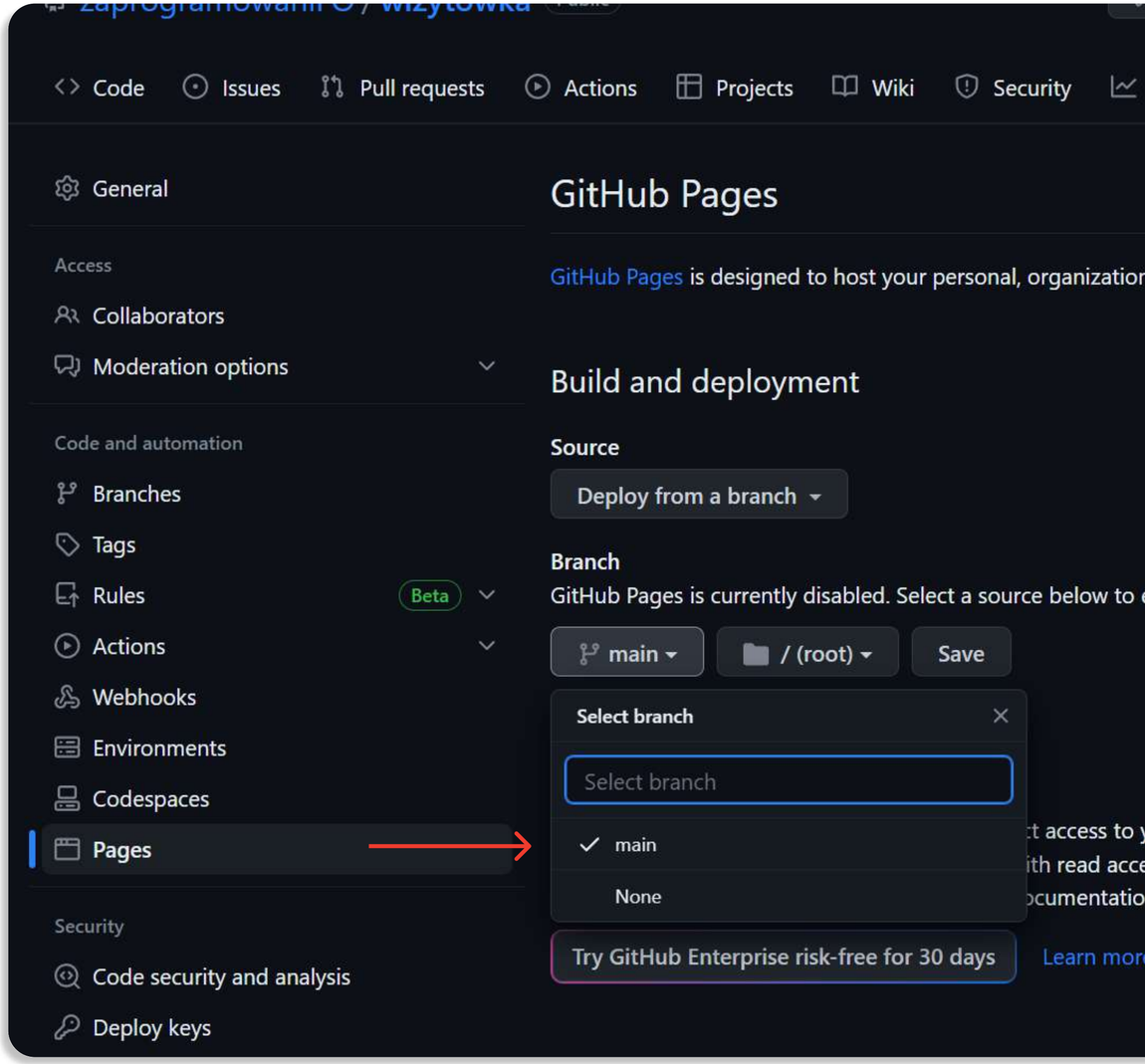


Publikujemy stronę

W sekcji **Branch** wybieramy **main**, a następnie klikamy **save**.

Po tym procesie nasza treść jest przetwarzana i po jakimś czasie pojawi się dostępna pod adresem

nazwa_użytkownika.github.io/nazwa_repozytorium /

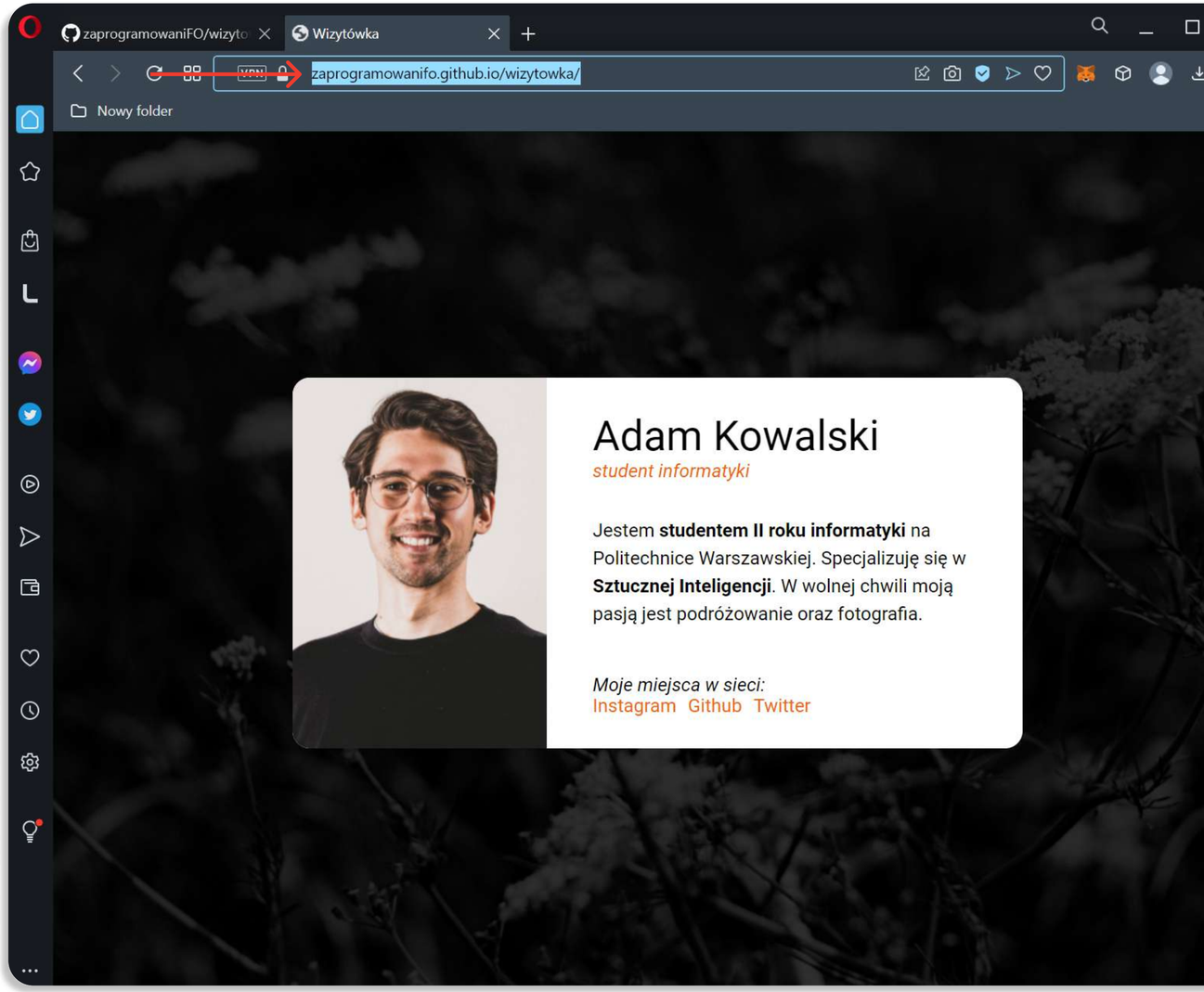


GOTOWE!

Strona opublikowana!

Uwaga!
Czasem przetworzenie danych, a zatem nim strona będzie widoczna w internecie, może zająć parę minut.

Cierpliwości! :)



</> ALGORYTMY

Czym są **algorytmy ?**

To skończony ciąg czynności, który umożliwia przekształcić dane wejściowe na dane wyjściowe;

To opis rozwiązanie danego problemu krok po kroku;

Etapy konstruowania algorytmu

1. Ustalenie problemu do rozwiązania dla algorytmu
2. Określenie danych wejściowych
3. Określenie wyniku oraz sposobu w jaki zostanie zaprezentowany
4. Wybranie metody wykonania zadania, która wg nas jest najlepsza
5. Zapisanie algorytmu
6. Sprawdzenie poprawności rozwiązania
7. Przeprowadzenie testów dla różnych danych wejściowych
8. Praktyczna ocena skuteczności algorytmu

PRZYKŁADEM ALGORYTMU MOŻE BYĆ PRZEPIS KULINARNY, NP. NA KANAPKĘ! DANYMI WEJŚCIOWYMI SĄ SKŁADNIKI, WYKONYWANYMI OPERACJAMI SĄ WSZYSTKIE CZYNNOŚCI W TRAKCIE PRZYGOTOWYWANIA KANAPKI, A DANYMI WYJŚCIOWYMI JEST GOTOWA KANAPKA! :)

Sposoby zapisania algorytmu

- opis słowny
- lista kroków
- pseudokod
- schemat blokowy - graficzna prezentacja algorytmu
- program komputerowy - zapis w danym języku programowania

</> RODZAJE ALGORYTMÓW

Algorytmy liniowe

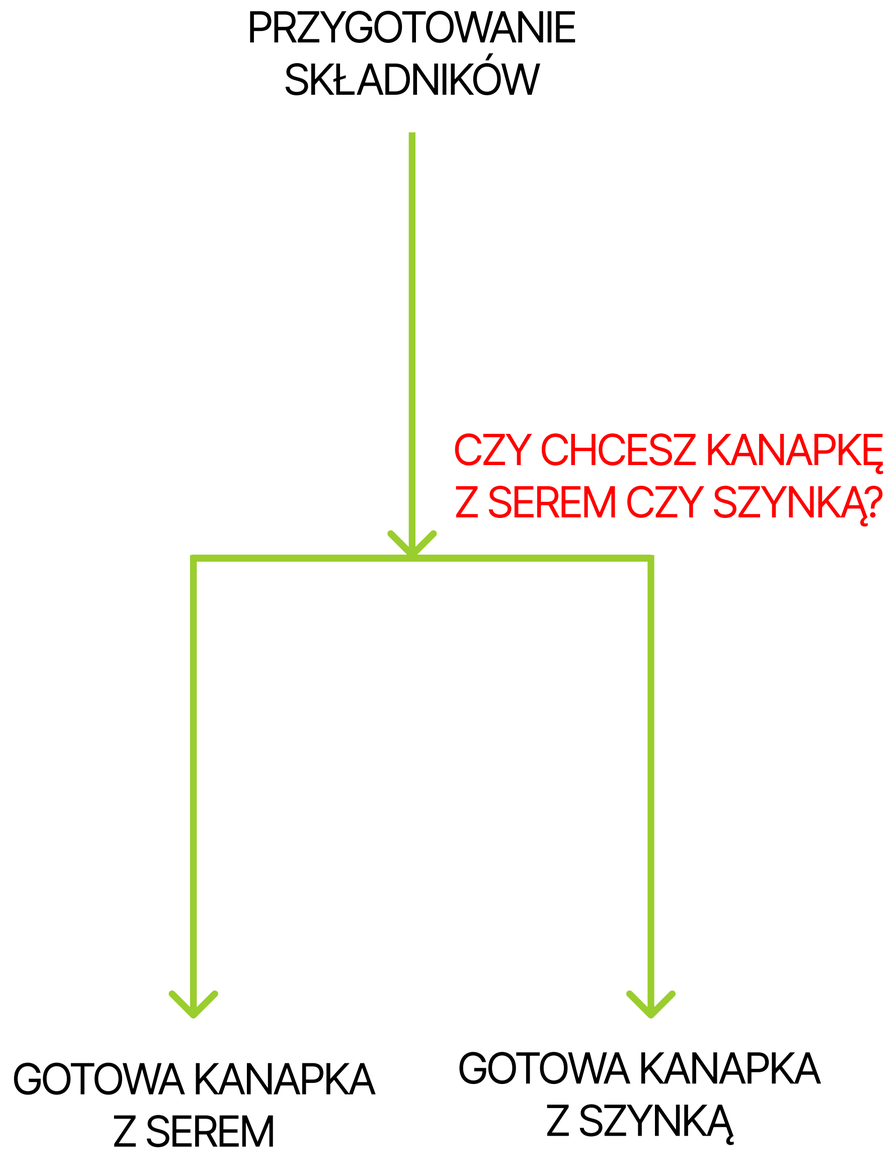
To takie algorytmy, w których nie określono żadnych warunków, a każde z poleceń wykonywane jest bezpośrednio jedno po drugim. Takie algorytmy nazywamy również jako sekwencyjne.

JEST TO NAJPROSTRZY ALGORYTM. ROZPATRUJĄC PRZYKŁAD KANAPKI, NA POCZĄTKU PRZYGOTOWUJEMY SKŁADNIKI, NASTĘPNIE ROBIMY KANAPKĘ I NA KOŃCU JĄ JEMY ;)

Algorytmy warunkowe

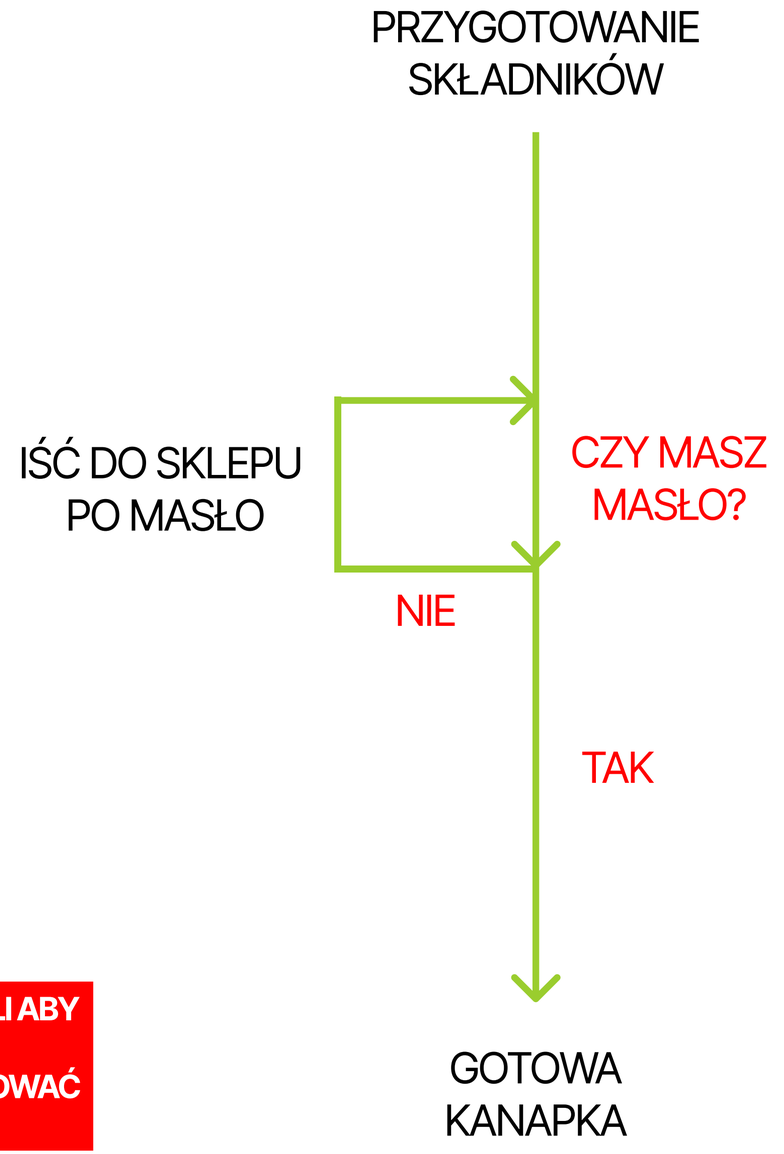
To algorytmy w których zostają postawione pewne warunki, i wykonanie konkretnych instrukcji jest uzależnione od ich spełnienia bądź nie.

ALGORYTM STAWIA JAKIŚ WARUNEK, PYTANIE KTÓRE W ZALEŻNOŚCI OD ODPOWIEDZI POPROWADZI NASTĘPNE DZIAŁANIA JEDNĄ Z DWÓCH MOŻLIWYCH ŚCIEŻEK.
W PRZYKŁADZIE KANAPKI, WARUNKIEM MOŻE BYĆ ZADECYDOWANIE CZY WOLIMY KANAPKĘ Z SZYNKĄ CZY Z SEREM.



Algorytmy iteracyjne

To algorytmy w których występuje powtarzanie pewnych operacji. Liczba powtórzeń może zostać z góry narzucona albo zależeć od spełnienia jakiegoś warunku, który za każdym powtórzeniem będzie sprawdzany.
Instrukcję powtarzania danego ciągu operacji nazywamy ITERACJĄ lub PĘTLĄ



POSTAWIONY WARUNEK, DOPÓKI NIE ZOSTANIE SPEŁNIONY, DOPÓTY NIE POZWOLI ABY WYKONANO KOLEJNE ZADANIA.
W PRZYKŁADZIE KANAPKI, DOPÓKI NIE MAMY MASŁA, TO NIE MOŻEMY PRZYGOTOWAĆ KANAPKI.

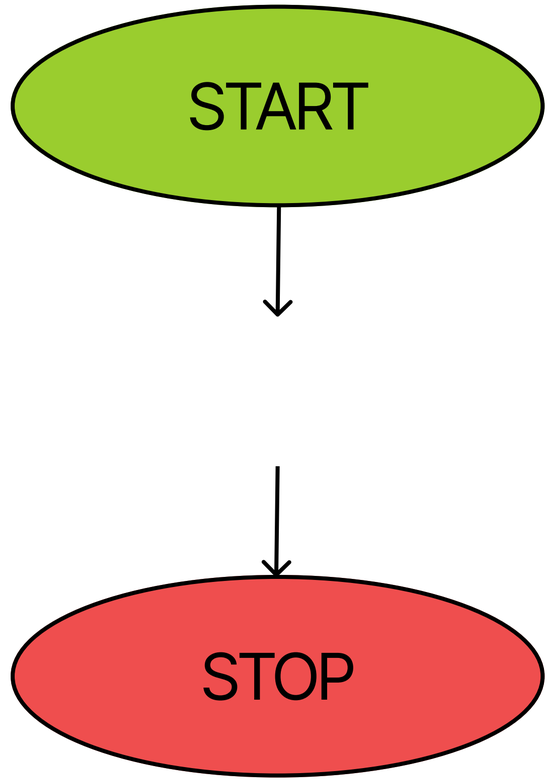
</> SCHEMATY BLOKOWE

Czym są schematy blokowe?

- Schematy blokowe umożliwiają przedstawienie algorytmu w postaci symboli graficznych, które opisują wszystkie wykonywane operacje w programie wraz z ich kolejnością.
- Pomagają zrozumieć problematykę danego zadania do rozwiązania.
- Pomagają w rozwiązaniu skomplikowanych zagadnień.

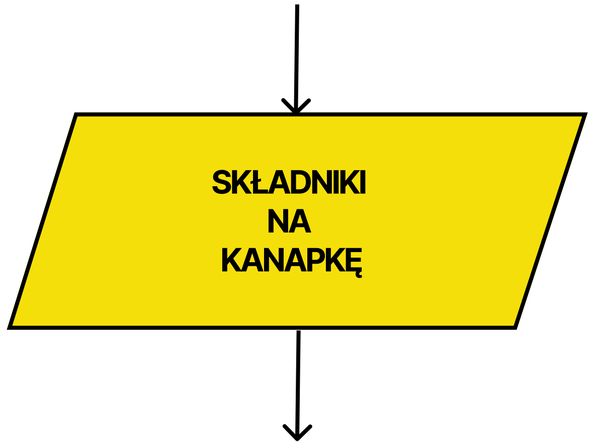
Bloki graniczne

- Bloki te oznaczają początek oraz koniec danego algorytmu. Mają kształt owalu.
- Każdy algorytm ma tylko jeden blok START, którego może wychodzić tylko jedno połączenie.
- Każdy algorytm może posiadać kilka bloków STOP (co najmniej jeden).



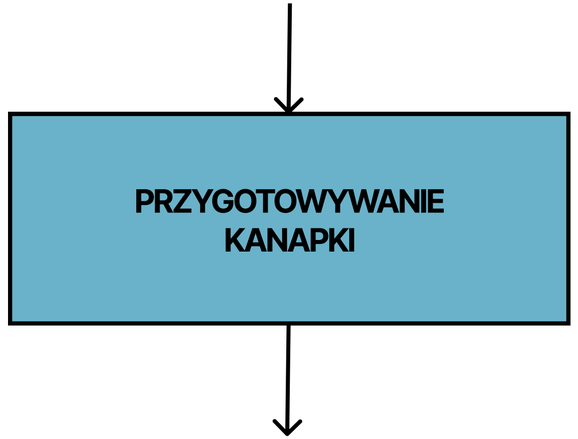
Blok wejścia / wyjścia

Blok ten odpowiada za wykonanie operacji wprowadzenia lub wyprowadzenia danych, wyników czy komunikatów.
Ma kształt podłużnego równoległoboku.
Może posiadać tylko jedno połączenie wejściowe oraz jedno połączenie wychodzące.
W jednym schemacie może znajdować się wiele takich bloków.



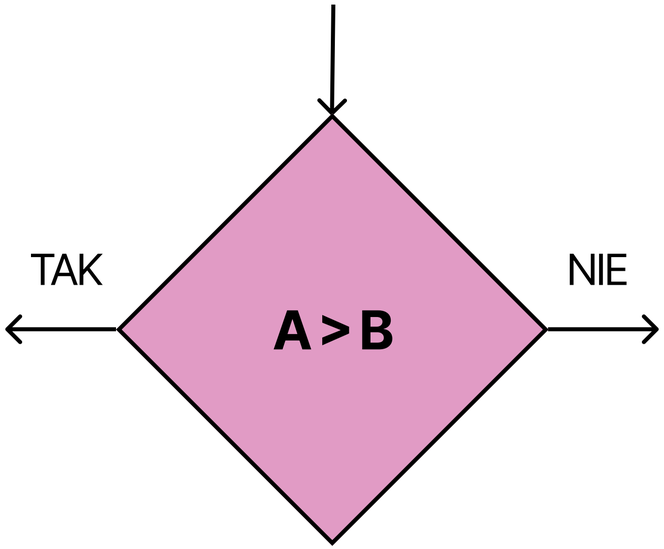
Blok operacyjny

Blok odpowiedzialny za wszelkie operacje w programie, które zmieniają wartości zmiennych, np. obliczenia.
Bloki operacyjne posiadają kształt prostokąta.
Jeden blok może zawierać więcej niż jedną operację.
Może posiadać tylko jedno połączenie wejściowe oraz jedno wychodzące.
W jednym schemacie może znajdować się wiele takich bloków.



Blok decyzyjny 1

Blok ten pozwala na wybranie jednej z dwóch możliwych ścieżek działania.
Ma kształt rombu.
Posiada tylko jedno wejście.
Posiada dwie ścieżki wychodzące:
TAK - jeśli wpisany warunek został spełniony
NIE - jeżeli wpisane warunek nie został spełniony

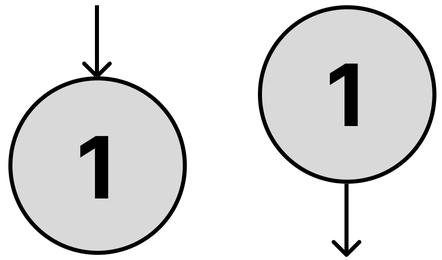


Połączenie bloków

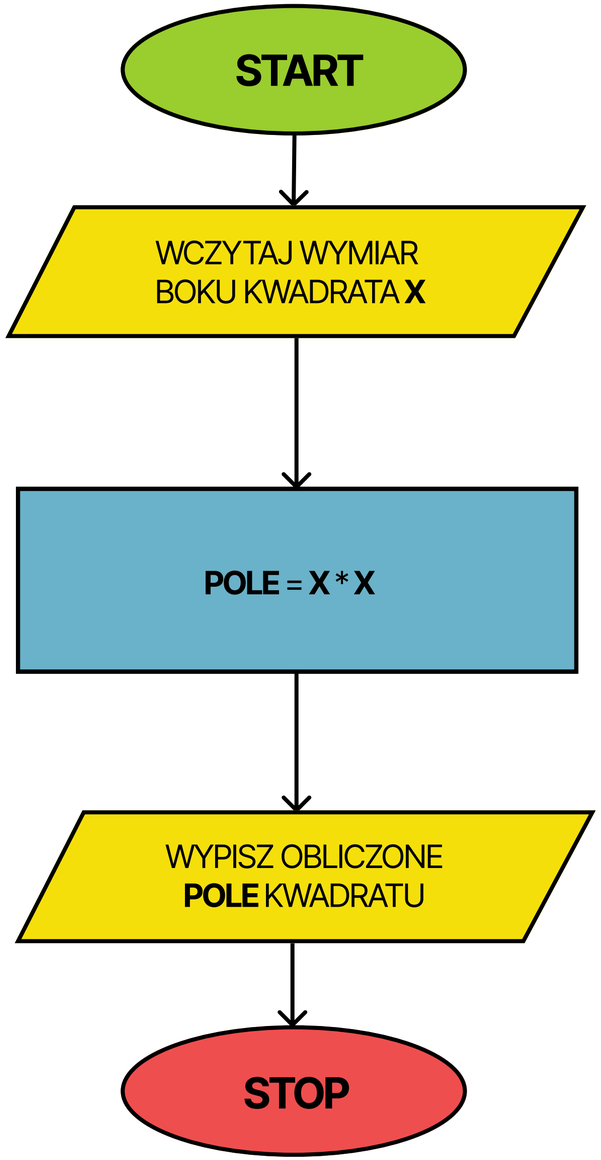
Połączenie odpowiedzialne jest za łączenie ze sobą bloków w schemacie. Określa kierunek w którą stronę przebiega przepływ danych czy też kolejność wykonywanych działań. Może też się łączyć z innymi połączeniami w jedno.

Łącznik schematów

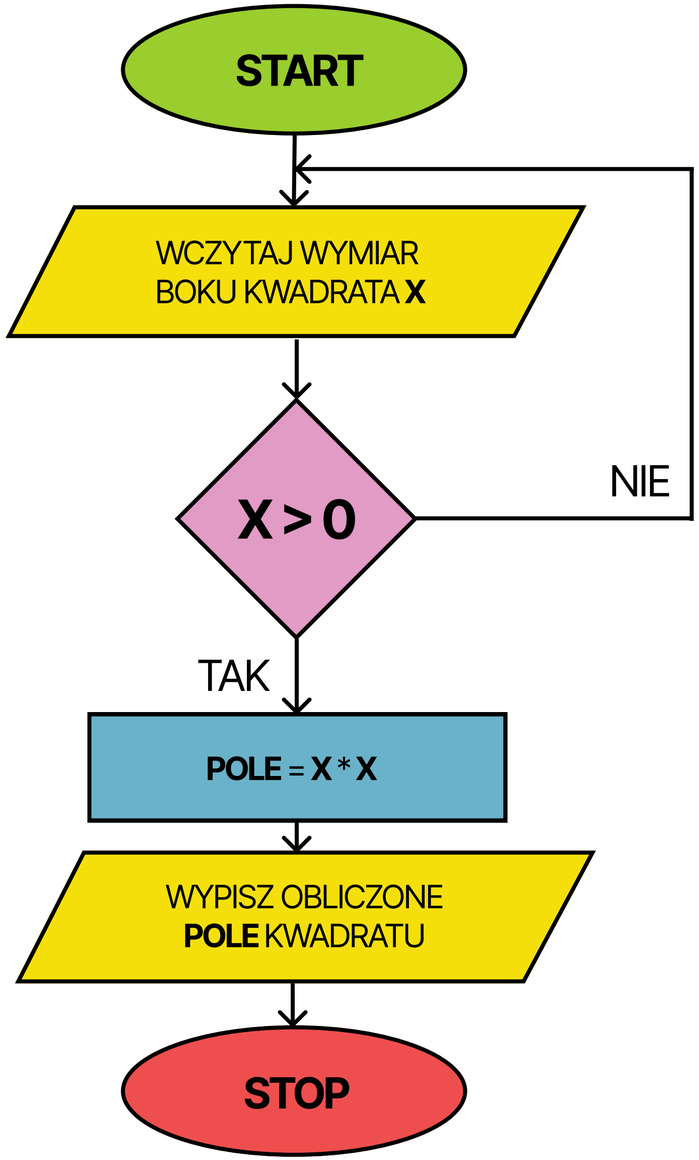
Stosujemy je, kiedy schemat blokowy chcemy narysować w kilku częściach. Jest to odsyłacz do innego fragmentu. Umieszczone oznaczenie wewnątrz musi być identyczne w obu częściach.

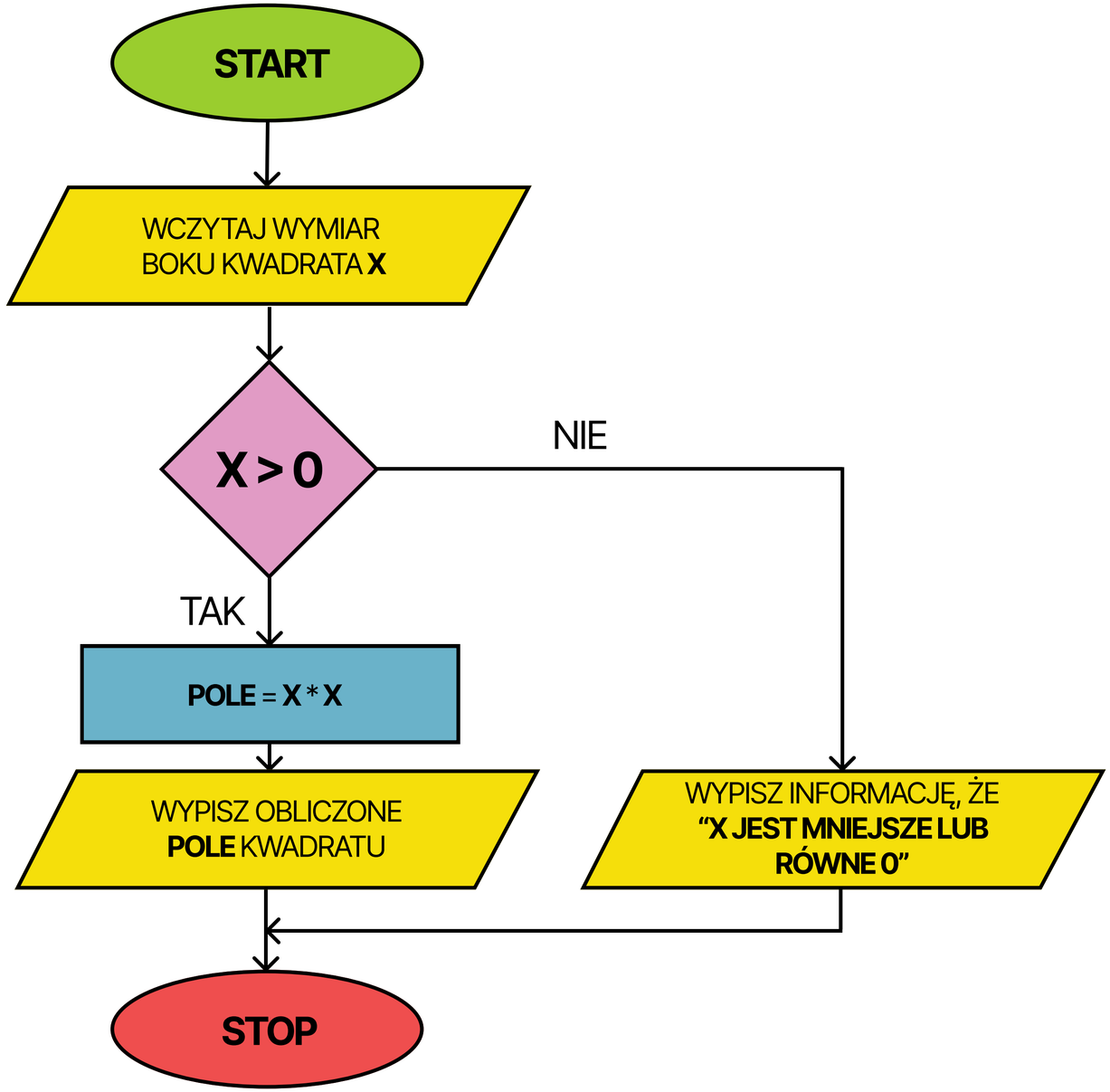


</> PRZYKŁADOWE SCHEMATY

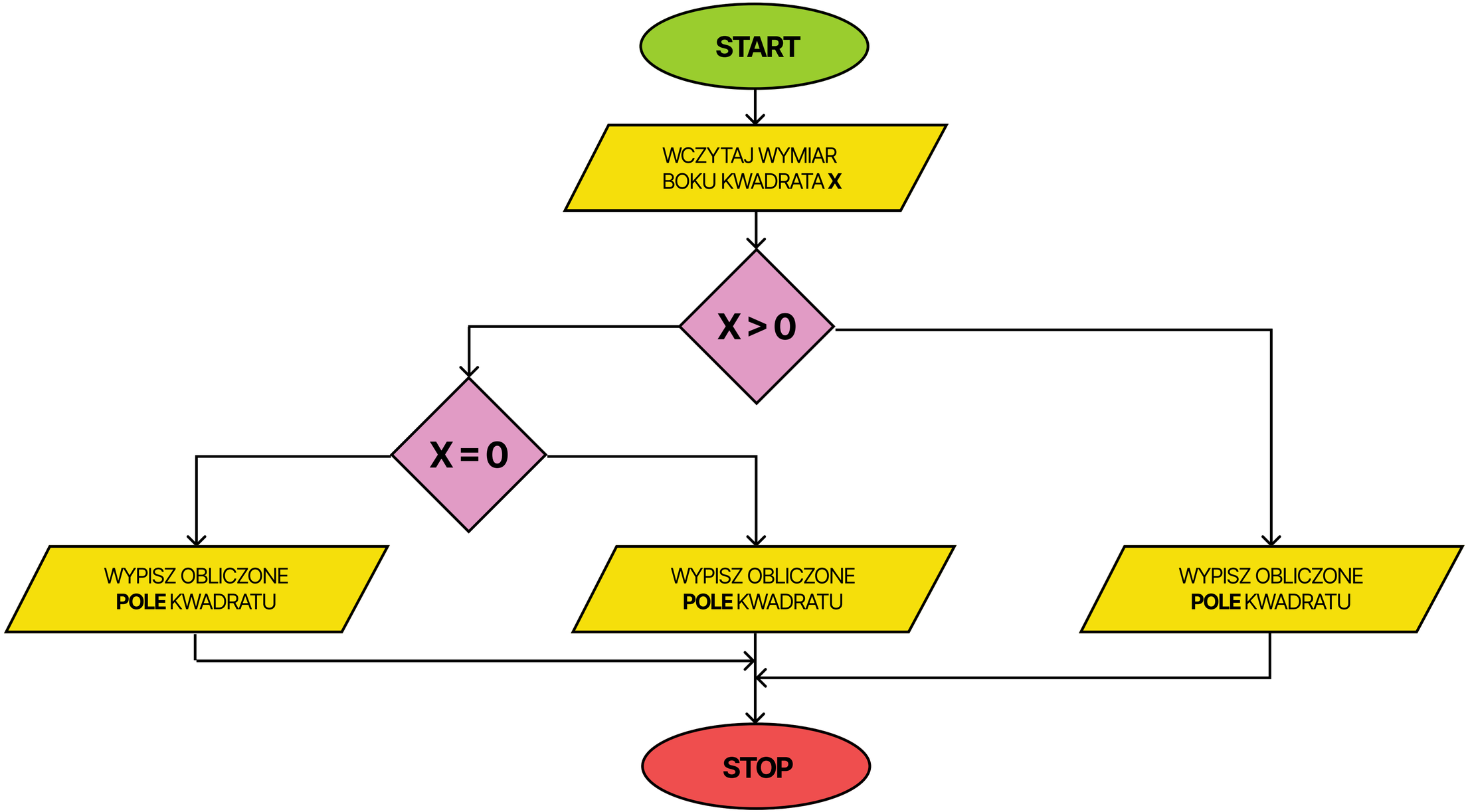


Algorytm na obliczenie pola kwadrata





Algorytm na obliczenie pola kwadratu, pod warunkiem boku większego niż 0



Algorytm sprawdzający czy X jest większy, mniejszy lub równy 0