

Dokumentacja

1.Temat: Kalkulator

Link: https://github.com/Gazda02/Kalkulator_gr2/tree/master

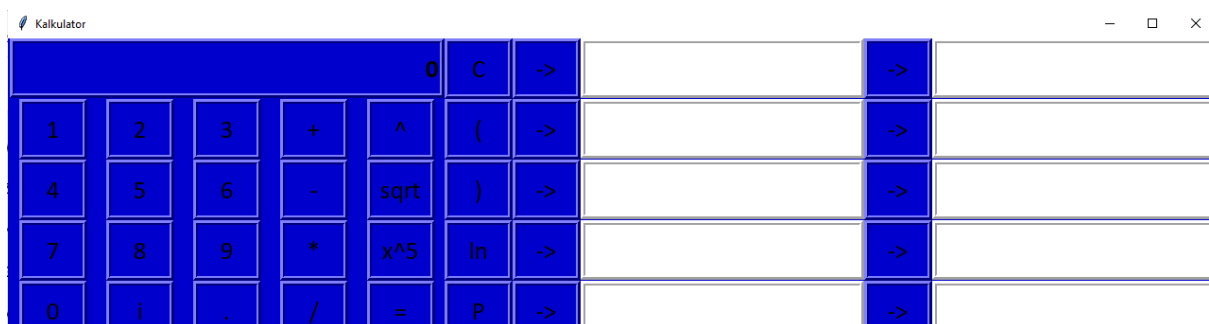
Grupa 3 zespół nr. 2:

1. Karol Gazda – Logika kalkulatora, łączenie funkcji stworzonych przez pozostałych członków zespołu, debugowanie
2. Weronika Juszczyk - Interfejs graficzny aplikacji, funkcja liczenia piątej potęgi
3. Piotr Grzyb Funkcjonalność pamięci ostatnich 10 zadań oraz odwoływanie się do zapamiętanego zadania oraz czyszczenie pamięci.
4. Jakub Gicala – Obsługa liczb zespolonych, dokumentacja

2.Funkcjonalności

1. Przeprowadzanie działań: dodawanie, odejmowanie, mnożenie, dzielenie, potęgowanie, pierwiastkowanie.
2. Pamięć ostatnich 10 zadań.
3. Odwołanie się do zapamiętanego zadania.
4. Czyszczenie pamięci.
5. Obsługa liczb zespolonych
6. Obliczanie piątej potęgi wybranej liczby
7. Interfejs Graficzny

Wygląd interfejsu aplikacji:



Program został stworzony z wykorzystaniem biblioteki tkinter, umożliwia on wprowadzanie i obliczanie prostych działań matematycznych oraz przechowywanie historii działań. Interfejs graficzny programu składa się z okna kalkulatora, którego wymiary można dostosować do indywidualnych potrzeb użytkownika. W oknie kalkulatora znajduje się pole do wprowadzania działań matematycznych oraz dziesięć pól, w których przechowywana jest historia ostatnich dziesięciu równań. Aplikacja została napisana przez kilku autorów. Każdy z nich odpowiada za określone funkcje programu. Autorzy wykorzystali bibliotekę math do obliczeń matematycznych. Funkcja "press" jest wywoływana po wciśnięciu przycisku i dodaje odpowiednią cyfrę lub operator do zmiennej "expression", która jest następnie wyświetlana na ekranie kalkulatora za pomocą funkcji "equation.set". Funkcja "clear" jest wywoływana po

naciśnięciu przycisku "C" i usuwa zawartość zmiennej "expression", a następnie wyświetla na ekranie kalkulatora wartość "0". Funkcja "press_equal" jest wywoływana po naciśnięciu przycisku "=" i próbuje obliczyć wartość równania zapisanego w zmiennej "expression" za pomocą funkcji "eval", która przyjmuje równanie jako argument i zwraca wynik. Wynik jest wyświetlany na ekranie kalkulatora za pomocą funkcji "equasion.set". Funkcja "clear_previous" jest wywoływana po naciśnięciu przycisku "CE" i usuwa historię wprowadzonych równań. Funkcja "previous_equasion" jest wywoływana po naciśnięciu przycisku "1-10" i wyświetla wprowadzone równanie z historii, której numer został podany przez użytkownika. Kod zawiera również definicje interfejsu użytkownika (GUI), w tym okna kalkulatora, ramek, pól tekstowych i przycisk. Każdy z członków zespołu odpowiada za inną funkcjonalność aplikacji. Program kalkulatora jest zatem efektem wspólnej pracy kilku programistów, którzy wykorzystali różne narzędzia i biblioteki, aby stworzyć przyjazny i funkcjonalny interfejs użytkownika.

3.Proces tworzenia projektu wraz z datami

- 14.03.2023 – Zebranie zespołu i ustalenie harmonogramu pracy – Cały zespół
- 15.03.2023 – Rozpisanie planu projektu z podziałem zadań – Jakub Gicala
- 19.03.2023 – Utworzenie podstawowej funkcjonalności kalkulatora (dodawanie, odejmowanie, mnożenie, dzielenie) – Karol Gazda
- 19.03.2023 – Stworzenie podstawowego GUI do testów – Weronika Juszczuk
- 19.03.2023 – Dodanie funkcjonalności potęgowania oraz pierwiastkowania – Karol Gazda
- 19.03.2023 – Dodanie obsługi pamięci ostatnich 10 zadań oraz odwoływanie się do zapamiętanego zadania – Piotr Grzyb
- 20.03.2023 – Dodanie funkcjonalności czyszczenia pamięci - Piotr Grzyb
- 20.03.2023 – Dodanie obsługi liczb zespolonych – Jakub Gicala
- 20.03.2023 – Implementacja finalnego GUI – Weronika Juszczuk
- 21.03.2023 – Testowanie i poprawki błędów – Karol Gazda
- 24.03.2023 – Dodanie funkcji liczenia piątej potęgi – Weronika Juszczuk
- 02.04 2023 – Dodanie komentarzy objaśniających działanie kodu – Piotr Grzyb
- 02.04.2023 – Stworzenie dokumentacji projektu – Jakub Gicala

4.Wnioski

Narzędzia użyte podczas tworzenia projektu

- Python 3 - to popularny język programowania, który jest używany w tworzeniu różnego rodzaju aplikacji i narzędzi, w tym kalkulatorów. Jego prostota, czytelność i elastyczność czynią go idealnym narzędziem do pracy w grupie nad projektem, takim jak kalkulator. Python 3 zapewnia wiele przydatnych funkcji, takich jak obsługa wyjątków, pętle i funkcje, które mogą być wykorzystane do stworzenia funkcjonalnego kalkulatora.
- PyCharm - to jedno z najpopularniejszych środowisk programistycznych dla języka Python. Zapewnia wiele przydatnych funkcji, takich jak debugger, narzędzia do testowania, automatyczną weryfikację składni, edycję kodu, refaktoryzację i wiele innych. Dzięki tym funkcjom, praca w grupie nad projektem kalkulatora w PyCharmie jest bardziej efektywna i produktywna.
- Tkinter - jest biblioteką języka Python, która umożliwia tworzenie interfejsów graficznych użytkownika (GUI). Tkinter zapewnia wiele elementów interfejsu użytkownika, takich jak przyciski, pola tekstowe, okna dialogowe i wiele innych, które mogą być wykorzystane do stworzenia intuicyjnego i funkcjonalnego interfejsu kalkulatora.
- GitHub i Git - to narzędzia kontroli wersji, które są używane do przechowywania i zarządzania kodem źródłowym projektu. GitHub umożliwia łatwe dzielenie się kodem między członkami grupy, a także udostępnianie kodu publicznie. Git zapewnia wiele funkcji, takich jak tworzenie gałęzi (branching), scalanie (merging), cofanie zmian (reverting) i wiele innych, co ułatwia pracę w grupie nad projektem kalkulatora i zapewnia skuteczną kontrolę nad kodem źródłowym.

Plusy pracy w grupie:

1. Dzielenie się pomysłami - Praca w grupie pozwala na wymianę pomysłów i spojrzenie na projekt z różnych perspektyw. Każdy z członków grupy może mieć własne pomysły na temat projektu, co może prowadzić do lepszych rozwiązań.
2. Podział obowiązków - Praca w grupie pozwala na podział zadań, dzięki czemu każdy może skupić się na swojej specjalizacji lub obszarze zainteresowań. Dzięki temu, cały projekt może zostać wykonany szybciej i bardziej efektywnie.
3. Współpraca - Praca w grupie wymaga interakcji i komunikacji między członkami, co prowadzi do lepszego zrozumienia projektu i lepszej koordynacji działań. Dzięki temu, każdy może wnieść swój wkład w projekt i poprawić jego jakość.
4. Możliwość uczenia się - Praca w grupie może pozwolić na zdobycie nowych umiejętności i wiedzy, dzięki temu, że członkowie grupy mogą uczyć się od siebie nawzajem. Można również wymieniać się wiedzą na temat języka Python i innych narzędzi wykorzystywanych w projekcie.

Minusy pracy w grupie:

1. Trudności z komunikacją - Praca w grupie wymaga dobrego kontaktu i komunikacji między członkami, a niekiedy może to być trudne, szczególnie gdy członkowie grupy mają różne podejścia i style pracy.
2. Trudności z koordynacją - W grupie może być trudno skoordynować działania i harmonogram pracy, szczególnie gdy członkowie grupy mają różne obowiązki i plany.
3. Różne poziomy zaangażowania - Członkowie grupy mogą mieć różny poziom zaangażowania w projekt, co może prowadzić do nierównomiernego rozkładu pracy i konfliktów.
4. Problemy z podziałem zadań - Praca w grupie wymaga dobrego podziału zadań i koordynacji działań, co może być trudne, szczególnie gdy członkowie grupy nie mają jasno określonych ról i odpowiedzialności.

Oto kilka pomysłów jak można ulepszyć nasz projekt:

- Dodanie funkcji takich jak funkcje trygonometryczne, logarytmy, funkcje logiczne, itp. Można również dodać obsługę równań lub funkcji.
- Dodanie większej ilości opcji zapisu zadań: Zamiast zapamiętywać tylko ostatnie 10 zadań, można dodać opcje zapisu do pliku lub bazy danych, co pozwoli na zachowanie większej ilości danych.
- Możliwość przeglądania historii zadań: Można dodać opcję przeglądania całej historii zadań, a nie tylko ostatnich 10.
- Ulepsz funkcjonalność czyszczenia pamięci: Zamiast prostego czyszczenia pamięci, można dodać opcję czyszczenia tylko wybranych zadań lub czyszczenia pamięci z określonego przedziału czasu.
- Dodanie opcję obliczania innych potęg: Można dodać opcję obliczania dowolnej potęgi.
- Można dodać więcej funkcji do interfejsu użytkownika, takich jak menu rozwijane, przyciski funkcyjne, kontrolki, itp. Aby interfejs był bardziej interaktywny i łatwiejszy w obsłudze.