

AGH University of Science
and Technology Krakow



Super Calculator
KALKULATOR DLA ELEKTRONIKÓW/INFORMATYKÓW
Projekt zaliczeniowy z przedmiotu
Języki Programowania Obiektowego [C++]

Autor: Mikołaj Wiśniewski

Kraków, 2024

Spis treści

1.	WSTĘP (INTRODUCTION).....	3
2.	FUNKCJONALNOŚĆ (FUNCTIONALITY).....	9
3.	ANALIZA PROBLEMU (PROBLEM ANALYSIS).....	9
4.	PROJEKT TECHNICZNY (TECHNICAL DESIGN).....	9
5.	OPIS REALIZACJI (IMPLEMENTATION REPORT).....	13
6.	OPIS WYKONANYCH TESTÓW (TESTING REPORT).....	14
7.	PODRĘCZNIK UŻYTKOWNIKA (USER'S MANUAL).....	15
8.	METODOLOGIA ROZWOJU I UTRZYMANIA SYSTEMU (SYSTEM MAINTENANCE AND DEPLOYMENT).....	17
	BIBLIOGRAPHY.....	10

Lista oznaczeń

API	Application Programming Interface
OOP	Object-Oriented Programming
SDK	Software Development Kit
STL	Standard Template Library
UML	Unified Modeling Language
MIT	Massachusetts Institute of Technology
GTest	Google Test
CLI	Command Line Interface
VCS	Version Control System
IDE	Integrated Development Environment
UI	User Interface

1. WSTĘP

Celem projektu "Super Calculator" jest stworzenie prostego kalkulatora dostępnego z poziomu CLI (konsoli), dedykowany dla elektroników i informatyków, oraz osób potrzebujących narzędzia do operowania między różnymi systemami liczbowymi. Kalkulator ten oferuje możliwość wykonywania podstawowych operacji matematycznych (**dodawanie, odejmowanie, mnożenie, dzielenie**) w różnych systemach liczbowych, takich jak **dziesiętny, binarny, ósemkowy i szesnastkowy**.

Projekt został wykonany w całości przeze mnie na potrzeby przedmiotu "Języki Programowania Obiektowego". Do napisania aplikacji wykorzystałem język C++ w standardzie 17, pracowałem natomiast na IDE (środowisk programistyczne) CLion 2021.2.2. Do pracy nad projektem podpiąłem swoje prywatne repozytorium w serwisie GitHub.

Przed rozpoczęciem projektu została opracowana owa dokumentacja według zaleceń, na podstawie której sporządzony został plan dalszego działania. Powyższa dokumentacja zawiera między innymi diagram UML, schemat USE CASE, oraz ustalono system zwinnej pracy AGILE, co sumarycznie pozwoliło na płynne przejście przez kolejne etapy rozwoju.

W celu zapewnienia wysokiej jakości kodu i pewności co do jego działania, został wykorzystany debugger do dokładnej analizy zachowań i przeprowadzenia testów jednostkowych. Do implementacji testów skorzystano z popularnej biblioteki Google Test (gtest), umożliwiającej sprawdzanie poprawności funkcji i metod naszej aplikacji. Dzięki tym narzędziom możliwe było skrupulatne sprawdzenie poprawności implementacji i eliminacja ewentualnych błędów na etapie rozwoju projektu.

Podstawowe założenia projektu:

1. Przygotowanie syntetycznego opisu wraz z dokumentacją.
2. Stworzenie osobistego repozytorium w GitHub.
3. Napisanie kodu w języku C++ spełniającego postawione funkcjonalności.
4. Wykorzystanie narzędzi: debugger, CMake, git.
5. Sprawdzenie poprawności działania kodu.
6. Testowanie oprogramowania z wykorzystaniem biblioteki GTest.

1. Wymagania systemowe (requirements)

- **Kompilator C++:** Aby zbudować i uruchomić projekt "Super Calculator", wymagane jest posiadanie kompilatora obsługującego standard C++17. W trakcie prac nad projektem używano kompilatora g++.
- **System kontroli wersji Git:** Repozytorium projektu znajduje się na platformie GitHub, co umożliwia łatwe zarządzanie kodem źródłowym, śledzenie zmian oraz współpracę zespołową. Instalacja Git jest zalecana w celu ułatwienia i przyspieszenia instalacji projektu i ewentualnego dalszego jego rozwoju.
- **Debugger (opcjonalnie):** W trakcie tworzenia projektu wykorzystywano debugger do identyfikacji i rozwiązywania błędów w kodzie. Możliwość korzystania z debuggera, dostępnego np. w środowisku CLion, jest zalecana dla programistów, aby ułatwić proces debugowania, jednak nie jest niezbędny do uruchomienia projektu.
- **CMake:** System CMake został użyty do konfiguracji procesu budowania projektu. Aby zbudować projekt, zaleca się posiadanie zainstalowanego narzędzia CMake w wersji umożliwiającej generację plików Makefile lub innych zależnych od systemu operacyjnego.
- **Biblioteka GTest:** W celu przeprowadzenia testów jednostkowych wykorzystano bibliotekę Google Test (GTest). Aby uruchomić testy, konieczne jest zainstalowanie i skonfigurowanie GTest w projekcie zgodnie ze schematem projektu.

Projekt "Super Calculator" był rozwijany i testowany na platformach z systemami operacyjnymi Windows. Zaleca się używanie środowiska programistycznego CLion do wygodnego zarządzania projektem, jednak projekt powinien być kompatybilny z innymi środowiskami programistycznymi obsługującymi język C++ np. Visual Studio Code.

2. Funkcjonalność (functionality)

- **Podstawowe Operacje Matematyczne:**

Kalkulator umożliwia 4 podstawowe operacje matematyczne względem dwóch liczb, niezależnie od systemu liczbowego, w którym są one reprezentowane:

- dodawanie
- odejmowanie
- mnożenie
- dzielenie

- **Obsługa Systemów Liczbowych:**

Główny system liczbowy, w którym są przeprowadzane wszystkie obliczenia. To na nim zbudowany jest silnik programu:

- dziesiętny

Kalkulator oferuje możliwość wprowadzania liczb w trzech innych systemach oraz wyświetlania w nich wyników wcześniejszych operacji:

- binarny
- ósemkowy
- szesnastkowy

- **Konwersje Liczb:**

Zastosowanie metod konwersji umożliwiających wprowadzanie liczb w jednym systemie i prezentowanie wyników w innym, dostarczając wrażenie, że kalkulator działa w wybranym systemie liczbowym.

- **Prosty Interfejs Użytkownika:**

Zapewnienie intuicyjnego interfejsu użytkownika, pozwalającego na wybór systemu liczbowego i wykonywanie podstawowych operacji matematycznych.

- **Przejrzysta Prezentacja Wyników:**

Wyniki operacji prezentowane są w czytelny sposób, z uwzględnieniem systemu liczbowego, w którym użytkownik pracuje.

3. Analiza problemu (problem analysis)

- **Praca w różnych systemach liczbowych**

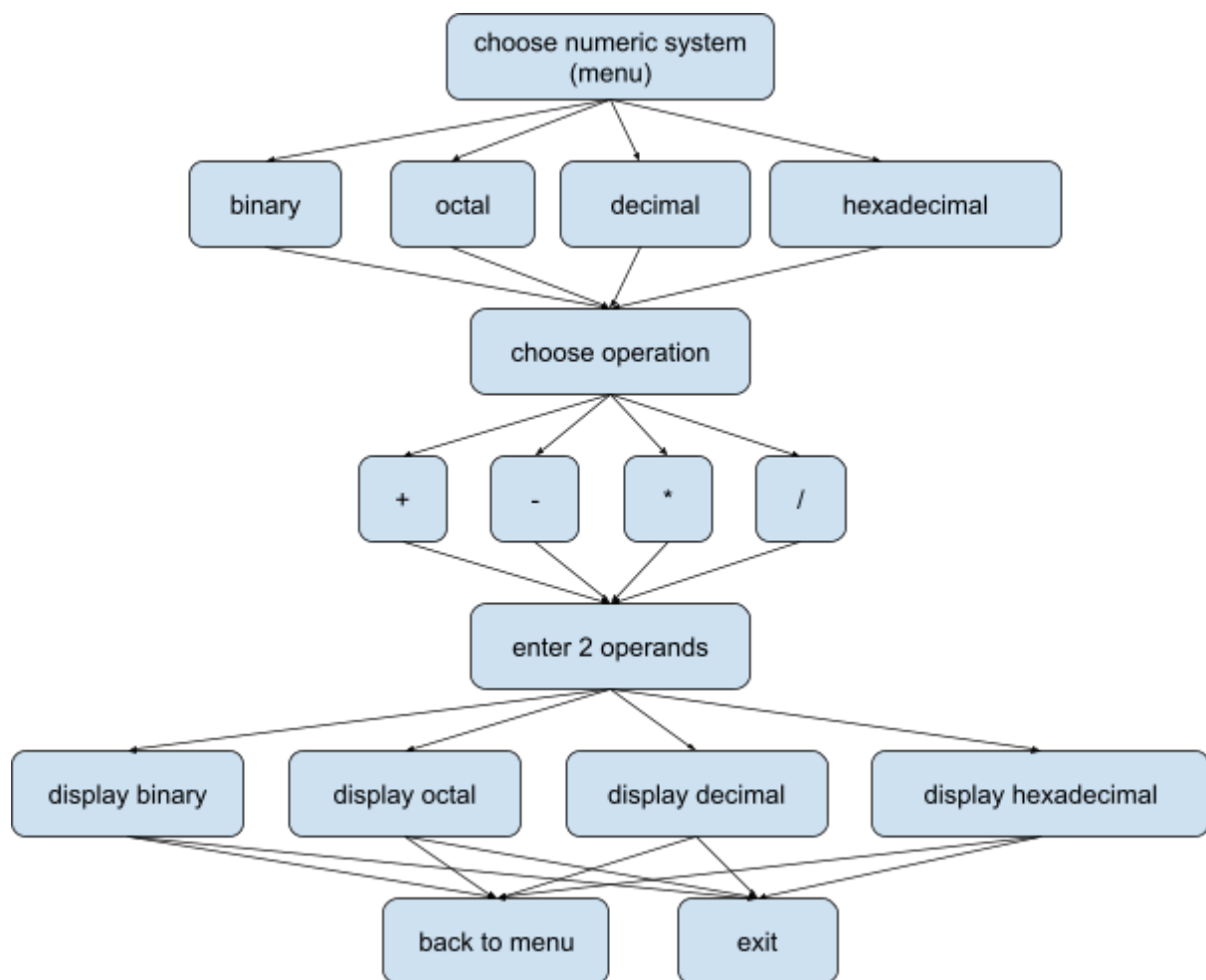
Elektronicy i informatycy często muszą operować na liczbach w różnych systemach liczbowych, takich jak binarny, ósemkowy, szesnastkowy oraz dziesiętny. Zrozumienie i wygodne korzystanie z tych systemów w celu dokonywania podstawowych operacji matematycznych jak i również zamiana między nimi jest kluczowa.

- **Dostępność z poziomu konsoli**

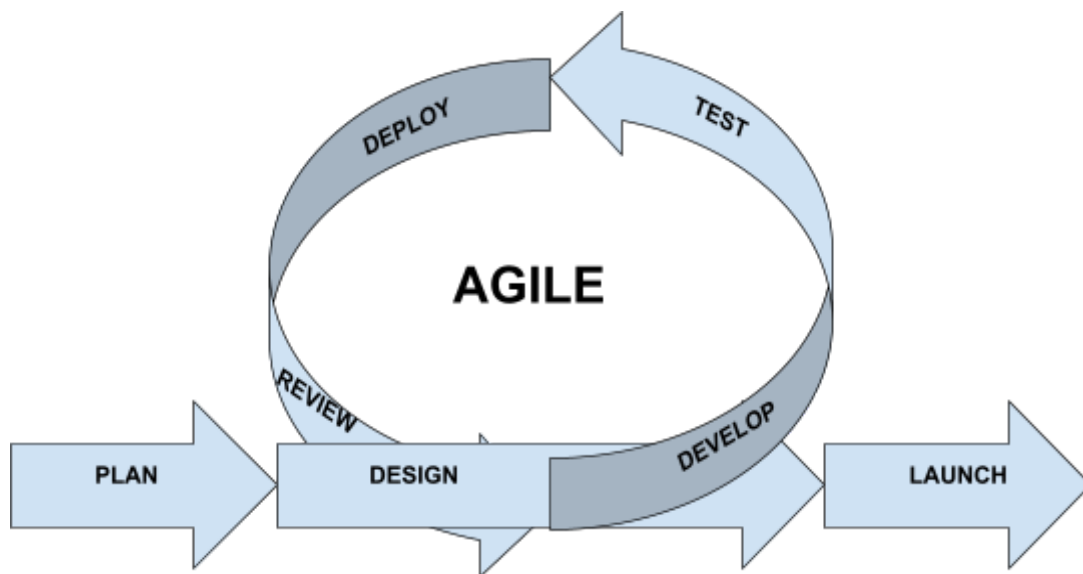
W praktyce często do podstawowych działań i funkcjonalności jak w przypadku tego projektu wygodniejsze dla użytkownika oraz prostsze w budowie są aplikacje konsolowe. Nie wymagają one dużych nakładów obliczeniowych maszyny oraz wiążą się z potencjalnie mniejszą ilością problemów jak w przypadku aplikacji z okienkowym GUI.

4. Projekt techniczny (technical design)

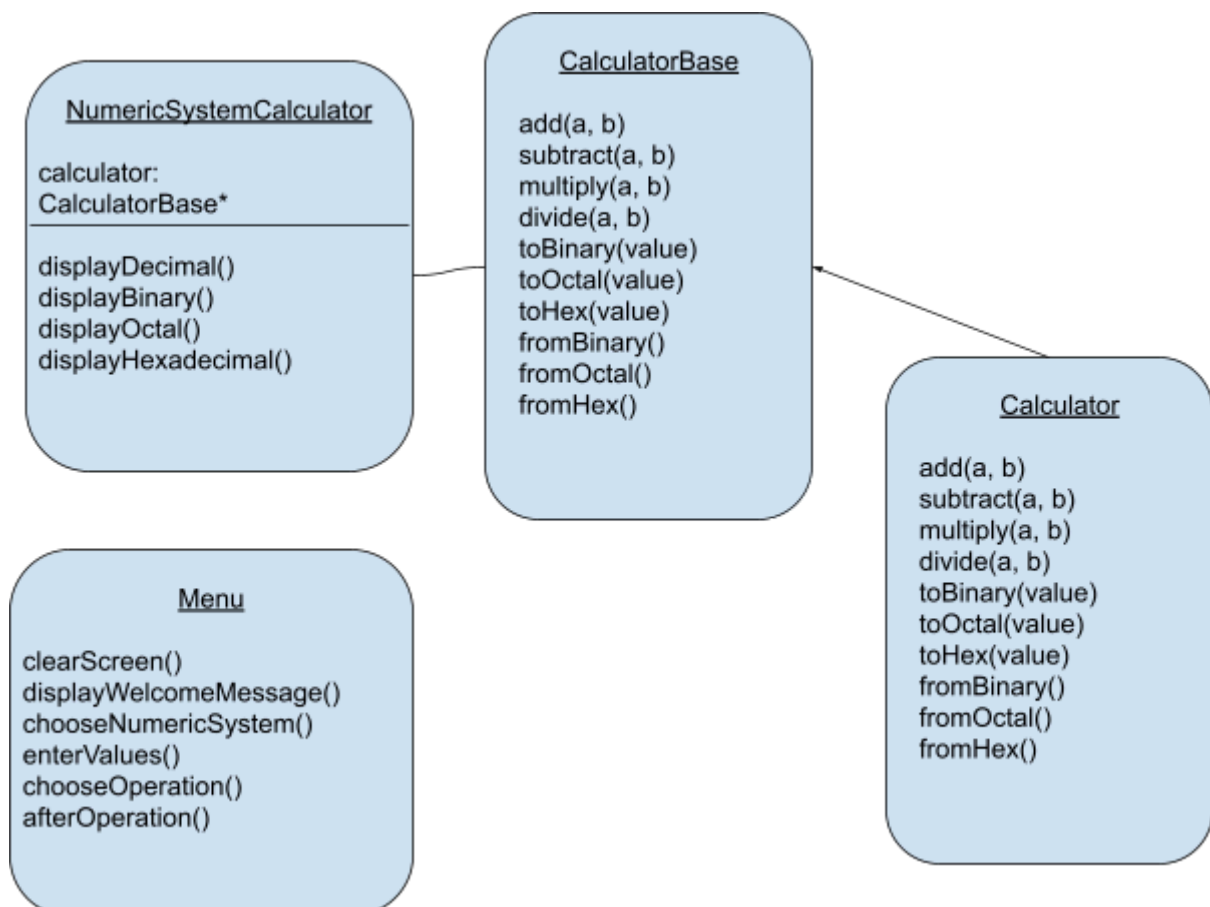
USE CASE



Agile Software Development Methodology



UML Class Diagram



5. Opis realizacji (implementation report)

- **Platforma testowa**

System Operacyjny: Windows 10 Home (08.02.2022)

Maszyna Testowa: komputer stacjonarny PC

Procesor: Intel(R) Core(TM) i5-3470 CPU 3.20GHz

Pamięć RAM: 12 GB DDR4

Dysk SSD: 250 GB

Dysk HDD: 500 GB

Kompilator [C++]: MinGW w wersji 8.1.0 dla systemu Windows.

Środowisko IDE: CLion 2021.2.2

- **Proces kompilacji i uruchomienia**

Klonowanie repozytorium

```
> git clone https://github.com/wisnuAGH/super_calculator.git
```

Przejdźcie do katalogu projektu

```
> cd super_calculator/cmake-build-debug
```

Kompilacja projektu

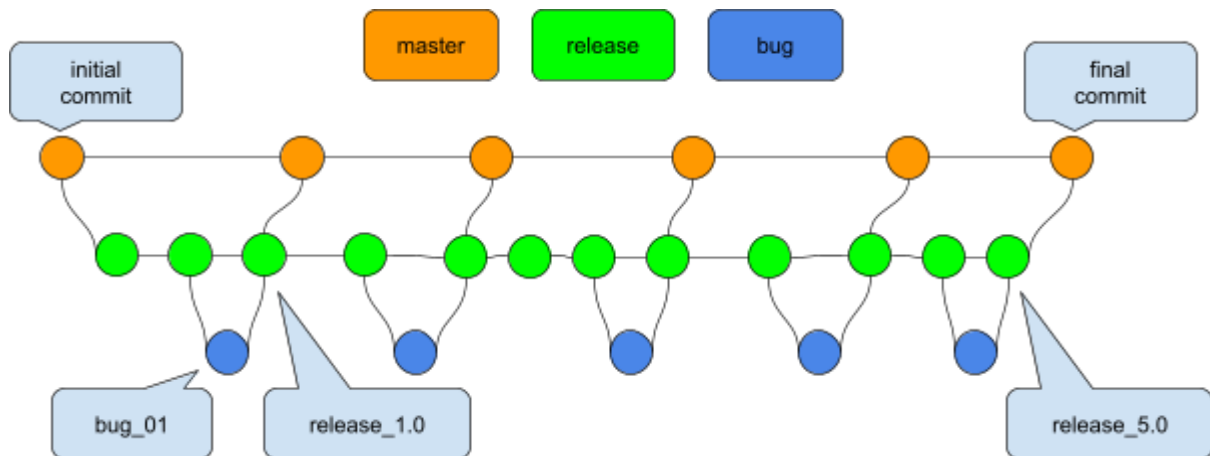
```
> make
```

uruchomienie aplikacji

```
> ./super_calculator.exe
```

- **System kontroli wersji**

Wykorzystany system kontroli wersji to Gi. Stworzono nowe repozytorium o nazwie “super_calculator” na prywatnym profilu w serwisie GitHub. Zastosowano hierarchię branchy dla 3 ścieżek (master - release_x.x - bug_xx)



- **Backups i zabezpieczenia**

Kopia Zapasowa Repozytorium - system kontroli wersji Git pozwala na skuteczną kontrolę historii i tworzenie kopii zapasowych repozytorium na platformie GitHub.

Częste Commity i Branching- częste commity umożliwiają śledzenie zmian w kodzie, a stosowanie branchingu pozwala na równoczesną pracę nad różnymi funkcjonalnościami bez ryzyka utraty danych.

Kontrola Dostępu do Repozytorium- repozytorium na platformie GitHub jest prywatne i zarządzane zgodnie z zasadami kontroli dostępu. W tym przypadku gdzie jest to projekt jednoosobowy, wszystkie operacje takie jak Merge Request czy integracja nowych zmian były przeprowadzane i kontrolowane przeze mnie osobiście.

Licencja MIT - Wykorzystanie licencji MIT gwarantuje pełną swobodę w korzystaniu, dostosowywaniu i dystrybuowaniu kodu źródłowego. Ten wybór stawia na otwartość i zachęca do aktywnego udziału społeczności programistycznej, stawiając na przejrzystość i elastyczność w korzystaniu z projektu.

6. Opis wykonanych testów (testing report) - lista buggów, uzupełnień, itd.

Kod usterki	Data	Autor	Opis	Stan
bug_01	14.01	Mikołaj Wiśniewski	- inability to switch between systems	RESOLVE
bug_02	14.01	Mikołaj Wiśniewski	- no possibility of repeating the program execution loop	RESOLVE
bug_03	14.01	Mikołaj Wiśniewski	- UI readability fixes	RESOLVE
bug_04	17.01	Mikołaj Wiśniewski	- display menu after even loop (1,3,5... loop display this)	RESOLVE
bug_05	17.01	Mikołaj Wiśniewski	- handling for bad value input	RESOLVE
bug_06	19.01	Mikołaj Wiśniewski	- expansion of input transforming functions	RESOLVE
bug_07	20.01	Mikołaj Wiśniewski	- need to add conversion option into UI	OPEN
bug_08	20.01	Mikołaj Wiśniewski	- handling for bad value input octal / hexadecimal (GTest)	OPEN

7. Podręcznik użytkownika (user's manual)

Super Calculator - Podręcznik Użytkownika

Spis Treści

1. Wstęp
2. Pobieranie Projektu
3. Konfiguracja
4. Uruchamianie Aplikacji
5. Korzystanie z Aplikacji
 - 5.1. Wybór Systemu Numerycznego
 - 5.2. Wprowadzanie Liczb
 - 5.3. Wybór Operacji
 - 5.4. Przegląd Wyników
 - 5.5. Kończenie Programu
6. Backupy i Zabezpieczenia
7. Licencja
8. Wsparcie Techniczne

1. Wstęp

Witaj w podręczniku użytkownika do Super Calculator! To proste narzędzie umożliwiające wykonywanie podstawowych operacji matematycznych oraz konwersji między różnymi systemami liczbowymi.

2. Pobieranie Projektu

Aby pobrać projekt, należy wykonać klon repozytorium:

```
> git clone https://github.com/wisnuAGH/super\_calculator.git
```

3. Konfiguracja

Projekt nie wymaga dodatkowych niestandardowych konfiguracji. Należy jednak upewnić się, czy na maszynie zainstalowany jest kompilator do C++ (np. g++) oraz narzędzie CMake.

4. Uruchamianie Aplikacji

Aby uruchomić aplikację, wykonaj następujące kroki:

Linux/macOS:

```
> cd super_calculator  
> g++ -o super_calculator main.cpp SuperCalculator.cpp  
> ./super_calculator
```

Windows:

```
> cd super_calculator  
> g++ -o super_calculator.exe main.cpp SuperCalculator.cpp  
> super_calculator.exe
```

5. Korzystanie z Aplikacji

5.1. Wybór Systemu Numerycznego

Po uruchomieniu pliku, wybierz preferowany system liczbowy, wpisując odpowiedni numer zgodnie z wyświetlanymi informacjami i naciskając Enter.

5.2. Wprowadzanie Liczb

Podczas wprowadzania liczb, postępuj zgodnie z instrukcjami na ekranie. Aplikacja obsługuje różne systemy liczbowe, więc dostosuj wprowadzane wartości do wybranego systemu wybierając wartość od 1 do 4.

5.3. Wybór Operacji

Po wprowadzeniu liczb, wybierz jedną operację matematyczną którą chcesz wykonać, wpisując odpowiedni numer w zakresie od 1 do 4 i naciskając Enter.

5.4. Przegląd Wyników

Wyniki operacji będą wyświetlane w wybranym systemie liczbowym. Możesz przeglądać wyniki zarówno w formie dziesiętnej, jak i w systemach binarnym, ósemkowym, oraz szesnastkowym.

5.5. Kończenie Programu

Aby zakończyć program, po zakończeniu operacji, naciśnij Enter. Program zapyta, czy chcesz kontynuować. Wybierz '1' i naciśnij Enter, aby kontynuować, lub inny klawisz, aby zakończyć.

6. Backupy i Zabezpieczenia

Projekt wykorzystuje licencję MIT, co oznacza swobodę w korzystaniu z kodu. Zalecamy regularne tworzenie kopii zapasowych repozytorium, aby zabezpieczyć się przed utratą danych.

7. Licencja

Projekt jest udostępniany na licencji MIT. Szczegóły znajdziesz w pliku LICENSE.md.

8. Wsparcie Techniczne

W przypadku problemów, pytań lub propozycji rozwinięcia programu skontaktuj się poprzez GitHub Issues. Życzę udanej pracy z Super Calculator!

8. Metodologia rozwoju i utrzymania systemu (system maintenance and deployment)

Projekt "Super Calculator" jest rozwijany zgodnie z autorską metodologią, która łączy w sobie elementy metodyki Agile-SCRUM oraz własne doświadczenie zdobyte na stażu jako software developer. Podejście to pozwala na elastyczne dostosowanie projektu do zmieniających się wymagań oraz umożliwia szybką reakcję na potencjalne błędy.

Główne Założenia Metodologii:

1. Samodzielny Rozwój:

Autorski charakter projektu sprawia, że każdy etap, począwszy od pisania kodu, poprzez debugowanie, aż po testy jednostkowe, jest wykonywany przeze mnie.

Samodzielna kontrola nad repozytorium na platformie GitHub, w tym zarządzanie gałęziami i wersjami.

2. Branching i Wersjonowanie:

Wersje systemu są wdrożone zgodnie z systemem semantycznego wersjonowania (SemVer), co ułatwia zrozumienie wprowadzanych zmian i aktualizacji.

Poszczególne etapy wdrożeń oznaczone są odpowiednimi branchami, takimi jak release_1.0, release_2.0, a błędy poprawiane są w gałęziach bug_01, bug_02, itd.

3. Zastosowanie Metodyki Agile-SCRUM:

Choć projekt nie jest dużą grupową inicjatywą, korzystanie z zasad Agile-SCRUM pozwala na efektywne planowanie i zarządzanie pracą.

Plan działania, dokumentacja projektu oraz struktura kodu zostały dostosowane zgodnie z zaleceniami tej metodyki.

4. Przygotowanie i Planowanie:

Przed przystąpieniem do projektu została opracowana dokumentacja według zaleceń, na podstawie której sporządzony został plan działania.

Realizacja projektu rozpoczęła się od starannego przygotowania dokumentacji oraz UML, co pozwoliło na bardziej płynne przejście przez kolejne etapy rozwoju.

Zarządzanie Repozytorium:

Każda zmiana w kodzie jest śledzona, a kontrola nad gałęziami, branchami oraz merge'ami jest samodzielnie przeprowadzana.

Przed wdrożeniem do gałęzi master autoryzowane są kolejne wersje, a wszelkie błędy (bugi) są aktywnie monitorowane i rozwiązywane przed zatwierdzeniem zmian.

Podsumowanie:

Zastosowana metodologia rozwoju i utrzymania systemu w projekcie "Super Calculator" łączy w sobie samodzielną inicjatywę autora z elementami metodyki Agile-SCRUM. To podejście umożliwia skuteczne zarządzanie procesem, szybką adaptację do nowych wyzwań oraz utrzymanie systemu na najwyższym poziomie funkcjonalności.

Bibliografia

- [1] Wykład oraz Laboratorium (prowadzący - prof dr hab. inż. Cyganek, B)
- [2] <https://github.com/>
- [3] <https://stackoverflow.com/>
- [4] <https://www.geeksforgeeks.org/>
- [5] <https://pl.wikipedia.org/wiki/>
- [6] <https://en.cppreference.com/w/>
- [7] <https://github.com/google/googletest>
- [8] <https://www.codingninjas.com/>
- [9] <https://www.w3schools.com/>
- [10] <https://www.programiz.com/>
- [11] <https://www.ibm.com/>
- [12] <https://docs.github.com/en/repositories/>

link do projektu w GitHub: https://github.com/wisnuAGH/super_calculator.git