

Politechnika Śląska w Gliwicach
Wydział Automatyki, Elektroniki i Informatyki

Projekt:
Programowanie Komputerów II

Temat projektu:

Kalkulator

Prowadzący projekt: Dr Inż. Tomasz Moroń

Wykonawca projektu : Tomasz Migala

Gliwice 2018/2019

1. Wstęp - Opis wykonywanego projektu:

Wykonywany przeze mnie projekt to Kalkulator. Jego głównymi zadaniami jest obliczanie podstawowych działań tj. dodawanie odejmowanie itd. ale i wykonywanie obliczeń na liczbach zespolonych, konwersji jednostek etc.

Projekt ten został wykonany przy użyciu zalecanej przez prowadzących zajęcia biblioteki QT.

W projekcie tym moim zadaniem było wykorzystanie polimorfizmu oraz wykorzystanie trzech z wielu elementów poznanych podczas laboratorium oraz wykładów. Polimorfizm zastosowałem dla klas Complex, oraz Tomek, których klasą bazową jest klasa Virtual. Te dwie klasy w projekcie wykorzystywane są jako tzw. QWidget'y, czyli dodatki do głównego okna QMainWindow. Natomiast elementy jakie zastosowałem to:

- Przeciążanie operatorów: Zastosowane w klasie Zespolona, która odpowiada za ogólne podejście do działań na liczbach zespolonych
Przykładowe przeciążenie:

```
Zespolona operator-(Zespolona b)
{
    return Zespolona(im - b.im, re - b.re);
}
```

- Strumienie pamięciowe: Zastosowane również w klasie Zespolona, w celu zwieracania obiektu zespolona, jako string.

```
std::string toString()
{
    std::ostringstream oss;
    oss << re << "+" << im;
    return oss.str();
}
```

- Kontenery STL: Zastosowałem std::vector w klasie Calculator w metodzie void pamięć() jako miejsce przechowywania liczb dodanych przez użytkownika do pamięci, oraz std::map, którą zastosowałem w klasie Tomek. W mapie pod kluczami odpowiednich jednostek przechowywane są odpowiednie wartości typu int wykorzystywane następnie w "Switch case".

2. Teoretyczne podstawy wykonywanego projektu:

Użyte przeze mnie struktury danych:

- std::map<QString, int> vec;

Mapy to zbiorowe kontenery przechowujące elementy danego typu, za pomocą kombinacji klucz-wartość zgodnie z określona kolejnością.

W powyższym przypadku zastosowałem kombinację QString - int -> czyli kluczem do elementu jest typ QString, a wartością kryjącą się pod danym kluczem jest typ int. Elementy mapy są zawsze sortowane względem klucza. Mapa ta jest zaimplementowana jako drzewo wyszukiwania binarnego.

- `std::vector<QString> tab;`

Wektory to “pojemniki” reprezentujące tablice, które zależnie od potrzeb zmieniają swój rozmiar. Wektory wykorzystują ciągłe lokalizacje do przechowywania swoich elementów. Czyli, że jest możliwość tak jak w tablicach wyszukiwanie elementów poprzez wskaźniki.

Użycie liczb zespolonych:

- Liczby zespolone - liczby będące elementami rozszerzenie ciała liczb rzeczywistych o jednostkę urojoną ‘i’ to znaczy pierwiastek wielomianu $x^2 + 1$
- Działania na liczbach zespolonych wykorzystane w projekcie:

- *Dodawanie:*

$$(a + bi) + (c + di) = (a + c) + (bi + di)$$

- *Odejmowanie:*

$$(a + bi) - (c + di) = (a - c) + (bi - di)$$

- *Mnożenie:*

$$(a + bi)(c + di) = ac - bd + adi + bci$$

- *Dzielenie:*

$$\frac{a+bi}{c+di} = \frac{(ac+bd)+(bc-ad)i}{c^2+d^2}$$

- *Moduł liczby zespolonej* $z = a + bi$

$$|z| = \sqrt{a^2 + b^2}$$

- *Argument liczby zespolonej* $z = a + bi$:

$$\arg z = \arctan\left(\frac{a}{b}\right)$$

3. Dokumentacja zewnętrzna - instrukcja dla użytkownika :

Po skompilowaniu projektu oczom użytkownika ukaże się poniższe okno:



1. Wykonywanie obliczeń:

W okienku tym znajdują się operacje klasycznego prostego kalkulatora z pamięcią. Aby dokonać podstawowych obliczeń (+, -, *, /) użytkownik należy wykonać następujące kroki -> Wyklikać potrzebną liczbę numer 1 przy użyciu myszy, lub wystukać liczbę poprzez klawiaturę. Następnie wybrać interesującą nas operację(tylko przy użyciu myszy) ‘+’ lub ‘-’ lub ‘*’ lub ‘/’. Po wybraniu jednej z operacji należy wpisać kolejną liczbę do pożądanych obliczeń. Następnie aby obliczenie zostało wykonane należy nacisnąć przycisk ‘=’.

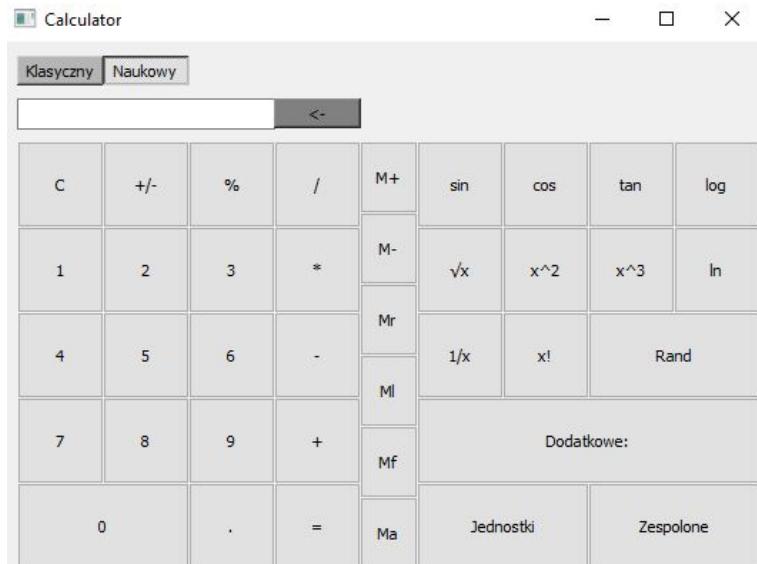
UWAGA: Obliczenie nie wykona się dopóki nie zostanie wciśnięty przycisk ‘=’ !!

2. Dodatkowe przyciski:

- **C** - po wciśnięciu tego przycisku obszar wynikowy zostanie skasowany i ustawiony na ‘0’
- **+/-** - Zmiana znaku wpisywanej liczby
- **%** - Przedstawienie liczby w procentach
- Przyciski odpowiadające za pamięć:
 - **M+** - Dodanie liczby znajdującej się w obszarze edycyjnym do pamięci
 - **M-** - Usunięcie z pamięci ostatnio dodanej liczby
 - **Mr** - Wyświetlenie w obszarze edycyjnym liczb z pamięci. Jeżeli liczb jest więcej niż jedna operacja ta wykona sumę wszystkich dodanych do pamięci liczb. Natomiast jeżeli użytkownik doda tylko jedną liczbę zostanie ona pokazana na ekran
 - **MI** - Wyświetlenie w obszarze edycyjnym ostatnio dodanej do pamięci liczby
 - **Mf** - Wyświetlenie w obszarze edycyjnym dodanej jako pierwszej do pamięci liczby
 - **Ma** - Wyświetlenie komunikatu który pokaże użytkownikowi jakie liczby dodał do pamięci podczas użytkowania kalkulatora od momentu komplikacji
 - “**<-**” - Usunięcie z obszaru edycyjnego ostatniego znaku

3. Zmiana na kalkulator naukowy:

Poprzez naciśnięcie przycisku “**Naukowy**” oczom użytkownika ukazuje się następujące okno:



1. Powrót do wersji klasycznej:

Użytkownik ma możliwość powrotu do okna klasycznego poprzez naciśnięcie przycisku “**Klasyczny**”

2. Przyciski naukowe:

- **sin, cos, tan** - obliczenie wartości trygonometrycznych liczby znajdującej się w obszarze edycyjnym.
- **log, ln** - obliczenie log(logarytm o podstawie 10), lub ln(logarytm naturalny) z liczby znajdującej się w obszarze edycyjnym
- **\sqrt{x} , x^2 , x^3 , $1/x$, $x!$** - (x - liczba znajdująca się w obszarze edycyjnym) - kolejno możliwość obliczania pierwiastka, kwadratu, sześcianu, odwrotności, silni z liczby x ;
- **Rand** - wylosowanie losowej liczby, która zostanie wpisana do obszaru edycyjnego
Każdorazowo w działaniach ukazanych w obszarze naukowym, nie ma konieczności naciskania przycisku ‘=’ obliczenia tutaj wykonywane są od razu po naciśnięciu odpowiedniego przycisku.

3. Dwa dodatkowe przyciski :

- **Jednostki** - po wciśnięciu tego przycisku ukaże nam się rozszerzenie całego okna o dodatkowy moduł umożliwiający użytkownikowi przeliczanie jednostek długości.

Użytkownik musi nacisnąć myszą na obszar znajdujący się najwyżej w tym module dzięki czemu poprzez klawiaturę

będzie mógł wpisywać liczbę którą chce poddać przeliczeniu. Dostępne są dwie rozwijane listy. Na liście wyższej użytkownik wybiera jednostkę w jakiej podaje liczbę. Natomiast w liście znajdującej się niżej możliwy jest wybór na jaką jednostkę kalkulator ma przeliczyć. W celu przeliczenia jednostki należy nacisnąć przycisk "**Przelicz**". W celu zamknięcia tego modułu należy nacisnąć przycisk "**Zamknij**"

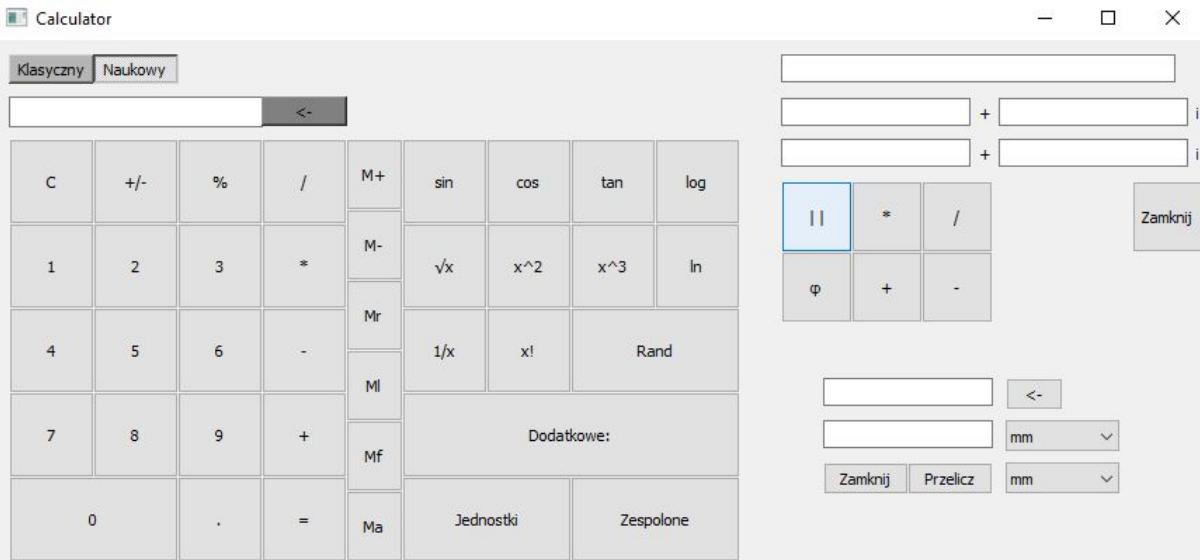
- **Zespolone** - po wciśnięciu tego przycisku ukaże nam się rozszerzenie całego okna o dodatkowy moduł umożliwiający użytkownikowi wykonywanie obliczeń na liczbach zespolonych.

The screenshot shows a user interface for complex number calculations. At the top, there are two input fields for complex numbers. Below them are four buttons: modulus (| |), multiplication (*), division (/), and addition (+). At the bottom, there are three buttons: argument (φ), subtraction (-), and a 'Zamknij' (Close) button. The entire interface is contained within a light gray box.

Najwyższy obszar służy tylko i wyłącznie do wyświetlania wyniku. W obszary edycyjne znajdujące się niżej użytkownik wpisuje liczby zespolone, które chce poddać operacjom matematycznym. Odpowiednio wierszami pierwsza i druga liczba zespolona, odpowiednio kolumnami część rzeczywista i część urojona.

- **| |** - Służy do obliczania modułu liczby zespolonej.
Jeżeli użytkownik wpisał dwie liczby zespolone moduły przedstawione są następująco:
1: | | - pierwszej 2: | | - drugiej
- **φ** - służy do obliczania argumentu liczby zespolonej.
Jeżeli użytkownik wpisał dwie liczby zespolone moduły przedstawione są następująco:
1: φ - pierwszej 2: φ - drugiej
- ***** - Wykonanie mnożenia dwóch liczb zespolonych
- **/** - Wykonanie dzielenia dwóch liczb zespolonych
- **+** - Wykonanie dodawania dwóch liczb zespolonych
- **-** - Wykonanie odejmowania dwóch liczb zespolonych
- **Zamknij** - Zamknięcie modułu liczb zespolonych

4. Kalkulator ze wszystkimi modułami:



4. Wnioski:

Na początku mojej przygody z programowaniem tj. na pierwszym semestrze, wydawało mi się, że programowanie to nie będzie dziedzina w której mógłbym znaleźć jakieś miejsce podchodziem do tego z dystansem, nie rozumiejąc zbyt wielu rzeczy raczej nie lubiałem tego przedmiotu, jednak wraz z tworzeniem tego projektu zrozumiałem wiele rzeczy, takie jak struktury danych, podejście obiektowe do programowania, przeciążanie operatorów i wiele innych, dzięki czemu programowanie stało się dla mnie bardziej przyjazne. Wyobrażenie o tworzeniu własnych projektów przy użyciu klawiatury bardzo nakręca u mnie chęć rozwijania się w tym kierunku.

Największą trudnością w moim projekcie było stworzenie dwóch dodatkowych modułów czyli konwertera jednostek oraz obliczania liczb zespolonych, w sensie odpowiedniego pokazywania tych modułów jako rozszerzenie głównego kalkulatora. Gdy udało mi się rozszerzać okno o te dodatki, następował problem z wciskaniem przycisków głównego kalkulatora mianowicie były one widoczne lecz nie dało się ich nacisnąć. Po wielu godzinach rozważania okazało się problem był trywialny tzn. okno które tworzyło się po prawej stronie było zbyt szerokie i przesłaniało główny kalkulator niewidocznym dla mnie oknem. W celu rozwiązania tego problemu użyłem metody dla dodatkowych modułów: ->setGeometry(), która przesunęła punkt zaczepienia lewego górnego rogu dodatku tak aby okna nie kolidowały ze sobą.