

ZAP II (projekt)

Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Spis treści

- 1. Temat projektu oraz jego krótki opis**
- 2. Założenia projektu**
- 3. Planowane działanie wersji alfa projektu**
- 4. Instrukcja korzystania z wersji alfa**
- 5. Opis działania poszczególnych modułów**
- 6. Planowane dodatki w wersji beta projektu**
- 7. Wprowadzone poprawki w wersji beta**
- 8. Założenia na wersję finalną**
- 9. Poprawki w wersji finalnej**
- 10. Przykładowe działanie programu**
- 11. Problemy, zabawne sytuacje oraz ogólne odczucia wobec projektu**

ZAP II (projekt)

Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Rozdział 1

Temat projektu oraz jego krótki opis

Tematem mojego projektu jest program mający za zadanie symboliczne wyznaczenie wzoru pochodnej funkcji wczytanej przez użytkownika.

ZAP II (projekt)

Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Rozdział 2

Założenia projektu

Głównym założeniem projektu jest stworzenie środowiska mającego za zadanie pomóc użytkownikowi z wyznaczeniem wzoru pochodnej wczytanej funkcji.

Pobocznymi założeniami są :

- **Modułowość** projektu
- **Zastosowanie** struktur dynamicznych
- **Korzystanie** z rekurencji

ZAP II (projekt)

Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Rozdział 3

Planowane działanie wersji alfa

Wersja alfa programu alfa powinna zawierać moduł mający za zadanie rozbić zadaną od użytkownika funkcji (zmiennej typu string) na strukturę dynamiczną w zależności od znaku (plus, minus, mnożenie,...) lub funkcji (na przykład sinus, logarytm naturalny). Strukturą dynamiczną będzie drzewo, w której rodzic będzie zawierał znak lub nazwę funkcji, a węzły będą zawierały część funkcji przed znakiem (lewy węzeł) oraz za znakiem (prawy węzeł). W przypadku funkcji wewnątrz będzie znajdowało się w lewym węźle.

Wersja alfa będzie ponadto zawierać moduł wyżej wymienionej struktury dynamicznej.

Pierwszy etap projektu powinien podawać wzory na pochodną podstawowych działań oraz kilku funkcji.

Wszystkie czynności będą odbywały się w konsoli.

Rozdział 4

Instrukcja korzystania z wersji alfa

1. Po otwarciu się okna konsolowego proszę wpisać funkcję, której pochodnej potrzebujemy. Proszę nie zapominać o znaku mnożenia w przypadku wpisywania wyrażeń (na przykład „2*x”), w przeciwnym wypadku program weźmie wyrażenie jako liczbę całkowitą.
2. Program wypisuje wzór na pochodną podstawowych działań oraz funkcji trygonometrycznych i logarytmu naturalnego. Poprawne ich zapisanie w konsoli przedstawia się w sposób następujący :
 - $\sin(\text{„tutajwpisujemyfunkcję”})$ dla sinusa
 - $\cos(\text{„tutajwpisujemyfunkcję”})$ dla cosinusa
 - $\text{tg}(\text{„tutajwpisujemyfunkcję”})$ dla tangensa
 - $\text{ctg}(\text{„tutajwpisujemyfunkcję”})$ dla cotangensa
 - $\ln(\text{„tutajwpisujemyfunkcję”})$ dla logarytmu naturalnego
3. W przypadku potęgowania program nie jest w stanie wypisać wzoru pochodnej jeśli funkcja musi być zamieniona na liczbę Eulera w potęgę.
4. Przykłady dobrze zapisanych funkcji :
 - $(\sin(2*x))^2/(3+x^3-\ln(x))-\text{tg}(x)$
 - $(22*x+\ln(\sin(x))-3)*(\text{tg}(x^2+\sin(x)))$
 - $2*x$
5. Wersja alfa programu wypisuje czysty wzór na pochodną, bez żadnych uproszczeń i czasem z nadmiarową ilością nawiasów.

Rozdział 5

Opis działania poszczególnych modułów

1. Drzewo

- **Moduł** odpowiedzialny za tworzenie, drukowanie oraz usuwanie dynamicznej struktury.

2. Rozbijanie

- **W** zasadzie najważniejszy moduł całego programu. Jego zadaniem jest rozbiecie wczytanej od użytkownika funkcji (zmiennej typu string) na drzewo. Funkcja działa od tyłu, usuwa końcowe nawiasy i dzieli string na części jeśli znajdzie operację matematyczną, bądź funkcję.

3. Pochodna

- **Funkcja** zamieniająca drzewo na pochodną i zwraca zmienną typu string. Do jej poprawnego działania potrzebny jest jeszcze moduł „zdrzewaznowunastringa”.

4. Zdrzewaznowunastringa

- **Funkcja**, która zamienia drzewo znowu na zmienną typu string, bardzo potrzebna w przypadku pochodnej mnożenia czy funkcji trygonometrycznych.

5. Upraszczenie

- **Zbiór** kilku funkcji, które w znaczącym stopniu upraszczają wypisaną wcześniejszą pochodną funkcji.

6. Interface

- **Wszystkie** wcześniejsze wywoływanie funkcji, czy tworzenie drzew i zmiennych będących wcześniej w „mainie” zostało przepisane do tej jednej funkcji.

ZAP II (projekt)

Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Rozdział 6

Planowane dodatki w wersji beta projektu

Wersja beta projektu zawierać będzie dodatkowy moduł o nazwie „**upraszczanie**”, którego zadaniem będzie uproszczenie pierwotnie podanego wzoru przez program.

Ponadto wersja beta będzie zawierała niewielkie zmiany w poprzednich modułach. Na przykład zamiana zmiennych typu string na zmienne liczbowe by poprawnie zapisać stopień potęgi, a następnie ponowna konwersja na zmienną typu string.

Wszystko nadal będzie odbywało się w oknie konsolowym.

ZAP II (projekt)

Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Rozdział 7

Wprowadzone poprawki w wersji beta

Dodanie modułu „upraszczanie” składający się z kilku mniejszych funkcji.

Wygląda w dość podobnie do „rozbijania” czy „zdrzewaznowunastringa”.

Znacząco uprasza wyrażenie podawane w wersji alfa programu.

Moduł „rozbijanie” wedle zaleceń został podzielony na kilka mniejszych funkcji i zbitý w jedną.

Usunięto w „drzewie” wyświetlanie go, które służyło tylko do sprawdzenia czy dobrze się tworzy.

Dodano moduł „interface”, który znacząco zmniejsza ilość linijek wpisanych w mainie.

ZAP II (projekt)

Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Rozdział 8

Założenia na wersję finalną

Obecność komentarzy w kodzie oraz pełna dokumentacja

Jeśli się uda – zamiana konsoli na GUI

ZAP II (projekt)

Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Rozdział 9

Poprawki w wersji finalnej

Poprawione zostało upraszczanie funkcji, a właściwie rozbijanie całej zmiennej typu string, brakowało dodania w każdej funkcji usuwania nawiasów.

Nie udało się niestety zaimplementować w wersji końcowej GUI, okazało się to być zbyt dużym wyzwaniem.

Dodano komentarze dla doxygena.

ZAP II (projekt)

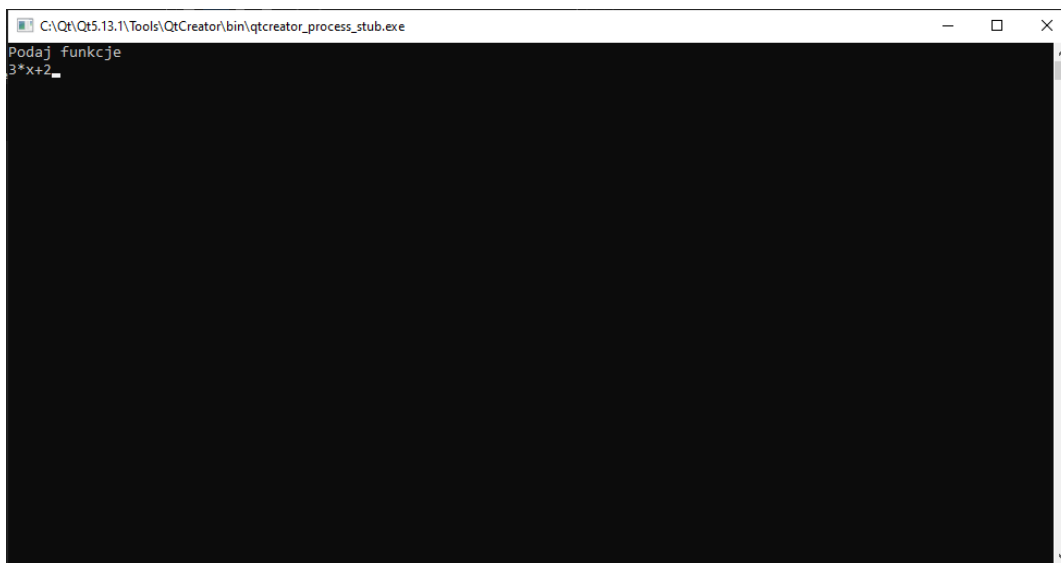
Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Rozdział 10

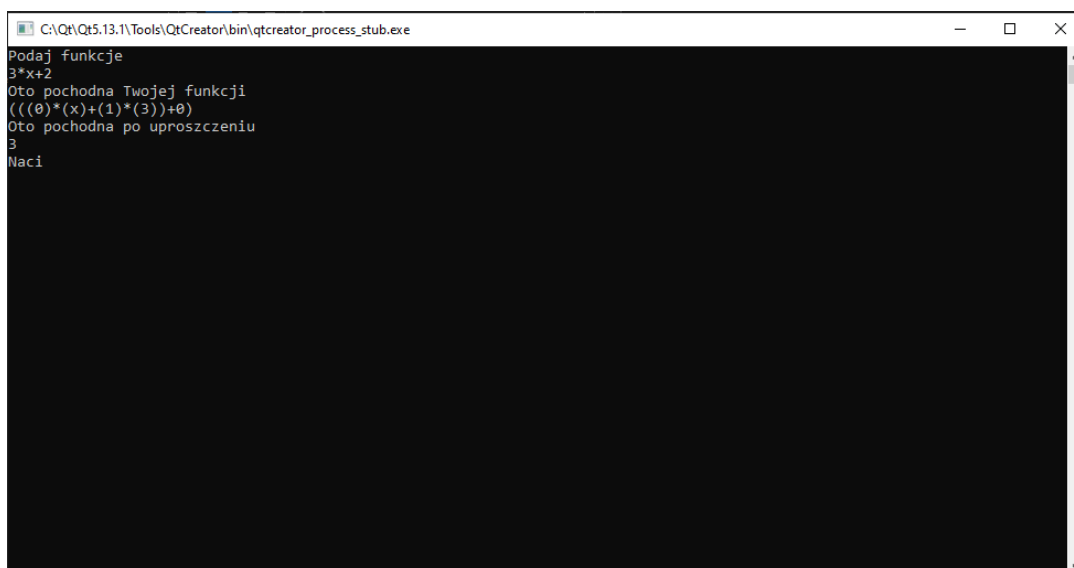
Przykładowe działanie programu

1. Wyświetla się konsola z tekstem „Podaj funkcję”
2. Należy wpisać funkcję



```
C:\Qt\Qt5.13.1\Tools\QtCreator\bin\qtcreator_process_stub.exe
Podaj funkcję
3*x+2
```

3. Klikamy „enter”, a programy wykonuje wcześniej zaprogramowany zestaw funkcji oraz wyświetla dwie rzeczy, pełną pochodną oraz niżej jej uproszczenie.



```
C:\Qt\Qt5.13.1\Tools\QtCreator\bin\qtcreator_process_stub.exe
Podaj funkcję
3*x+2
Oto pochodna Twojej funkcji
(((0)*(x)+(1)*(3))+0)
Oto pochodna po uproszczeniu
3
Naci
```

ZAP II (projekt)

Autor : Jakub Kozłowski

Opiekun : dr inż. Jan Klimaszewski

Rozdział 11

Problemy, zabawne sytuacje oraz ogólne odczucia wobec projektu

Ogólnie rzecz biorąc cały projekt pisało się bardzo przyjemnie, mimo kilku problemów, o których napiszę dalej.

Wersję alfa napisałem jakoś w 9h, usiadłem w pewien czwartek o 12 do komputera i odszedłem od niego jakoś koło 21 orientując się, że cały dzień zmarnowałem na pisanie i nawet nie zdążyłem zjeść obiadu. Klepanie kodu jednak wciąga.

Podobnie było z betą, jedno popołudnie i beta była napisana.

Niestety okazuje się, że środowisku qt, lubi płać figle amatorom programowania. Przeskakując między programami jak ja robiłem to wcześniej, czyli prawy->ustaw jako domyślny projekt, qt nie buduje od nowa programu. Przez to nie zauważyłem, że poprawki wprowadzone przeze mnie nie działają poprawnie.

Największym wyzwaniem okazało się, skrócenie najważniejszej funkcji jakim jest „rozbijanie”. W wersji beta zapomniałem na początku każdej mniejszej funkcji usuwać nawiasów przez co, upraszczanie często gubiło się, lub podawało głupie wyniki.

Komentowanie dla doxygena też było dość żmudne, cieszę się jednak, że do późnej bety ilość komentarzy w moim projekcie ograniczyłem do minimum. Niemniej jednak ilość komentarzy jakie musiałem napisać stanowiło większą część całej wersji finalnej projektu.