

Projekt: Kalkulator

Przedmiot:	Programowanie obiektowe
Imię i nazwisko autora:	Andrzej Morawski
Nr indeksu:	323455
Semestr studiów:	2
Data ukończenia pracy:	Czerwiec 2021 r.
Prowadzący pracownię:	Mgr Robert Kraus

Spis treści

Opis projektu	2
Wzorce projektowe.....	2
Zaimplementowane:	2
Możliwe do implementacji:.....	3
Opis działania aplikacji:	4
Źródła	7

Opis projektu

Projekt Kalkulator jest projektem końcowym zrealizowanym w ramach przedmiotu Programowanie Obiektowe. Celem projektu jest implementacja kalkulatora, który będzie imitował rozwiązanie zastosowane w systemie Windows 10 oraz będzie łatwy w obsłudze dla użytkownika tej aplikacji. Projekt został zrealizowany przy użyciu języka Python oraz bibliotek dostępnych dla tego języka takich jak PyQt5 czy sys.

Wzorce projektowe

Zaimplementowane:

W trakcie projektowania i implementacji kodu udało mi się kilka wzorców projektowych. Użycie ich znacząco ułatwia organizację pracy nad rozwijaniem aplikacji, oraz przyspiesza cały proces.

MVC (Model-View-Controller) to wzorec projektowy stosowany podczas projektowania aplikacji zawierających interfejs graficzny. Pozwala on na podzielenie programu na niezależne od siebie części, które będą realizowały inne zadania.

Model jest odpowiedzialny za wykonywanie określonych działań na danych, które są do niego przekazywane z aplikacji za pomocą kontrolera.

Widok jest to interfejs użytkownika, który pozwala na prostą interakcję z aplikacją.

Kontroler to logika odpowiedzialna za sterowanie aplikacją oraz przekazywanie danych z modelu do widoku i w drugą stronę.

W moim projekcie udało mi się zaimplementować MVC przez podział całej aplikacji na dwa główne rodzaje klas. Pierwszą odpowiedzialną za widok oraz kontrolowanie aplikacji np. Nazwa_GUI oraz drugą, która zajmuje się tylko działaniem na modelu np. Nazwa_Expression. Przykładowo w klasie Standard_GUI główną funkcją odpowiedzialną za

reprezentację widoku jest `initUI()`, która zawiera informację o wyglądzie aplikacji oraz pozwala na szybkie modyfikowanie jej. Funkcję kontrolera głównie realizuje metoda `clicked_button_expression()`, która jest odpowiedzialna za przekazywanie danych pomiędzy widokiem a modelem. Klasa `Standard_Expression` jest odpowiedzialna za realizację modelu. Za jej pomocą wyrażenie jest przekształcane do postaci gotowej do ewaluacji a następnie za pomocą wbudowanej w język Python funkcji `eval()` obliczana jest wartość tego wyrażenia. Następnie gotowy wynik jest zwracany do kontrolera, który jest odpowiedzialny za przekazanie tego wyniku do widoku.

Fasada jest wzorcem, którego głównym celem uporządkowanie systemu w taki sposób, aby móc łatwo ustalić dostęp użytkownika do poszczególnych podsystemów oraz ograniczyć dostęp do tych, które mogą zawierać chronione informacje.

Ze względu na nieduży rozmiar projektu ciężko jest wskazać rozbudowany przykład fasady, natomiast podstawowe założenia tego wzorca mogą być spełniane przez połączenia pomiędzy klasami odpowiedzialnymi za GUI oraz klasami odpowiedzialnymi za ewaluację wyrażenia, do których użytkownik nie ma bezpośredniego dostępu podczas korzystania z aplikacji.

Możliwe do implementacji:

Pamiętka jest czynnościowym wzorcem projektowym, którego zadaniem jest zapamiętanie oraz udostępnienie na zewnątrz stanu danego obiektu bez naruszania jego hermetyzacji.

Aby zaimplementować taki wzorec w projekcie można stworzyć nową klasę, która byłaby odpowiedzialna za przechowywanie zmiennych i w momencie otwierania nowego okna kalkulatora przekazywałaby do konstruktora informacje o zmiennych, które zawierała poprzednia klasa np. wartość ANS, wyrażenie, czy wynik działania.

Opis działania aplikacji:

Widok interfejsu graficznego po uruchomieniu programu.

Scientific		Programmer		Converter	
<input type="text"/>					
%	CE	C	DEL		
1/x	x^2	√x	÷		
7	8	9	×		
4	5	6	-		
1	2	3	+		
ANS	0	,	=		

Przyciski „Scientific”, „Programmer”, „Converter” są odpowiedzialne za przełączanie się pomiędzy pozostałymi oknami aplikacji, które zapewniają dodatkowe funkcjonalności. Aby móc wykonywać działania wystarczy za pomocą klawiszy wpisać wyrażenie a następnie nacisnąć przycisk „=”.

Poprawne wyrażenie np. $2+8^2-6$ zwróci wartość liczbową 60, natomiast wprowadzenie do kalkulatora błędnego wyrażenia np. $2+8^2-$ spowoduje zwrócenie wartości „ERROR”.

Działania $1/x$, x^2 czy \sqrt{x} działają na zasadzie szablonu co ułatwia użytkownikowi wprowadzanie danych do aplikacji. Natomiast przyciski „CE”, „C” i „DEL” umożliwiają edycję wprowadzonego działania. „CE” powoduje całkowite wyczyszczenie pamięci kalkulatora, „C” usuwa wyrażenie znajdujące się w okienku a operacja „DEL” pozwala na edycję pojedynczych znaków. Za pomocą przycisku „ANS” użytkownik jest w stanie wypisać na ekran ostatnią wyliczoną wartość.

Widok po przełączeniu kalkulatora na naukowy.

Standard					Programmer					Converter				
Switch	n	e	C	DEL	Switch	n	e	C	DEL	Switch	n	e	C	DEL
x^3	1/x	x	CE	%	x^2	1/x	x	CE	%	x^2	1/x	x	CE	%
$3\sqrt{x}$	()	_	÷	\sqrt{x}	()	_	÷	\sqrt{x}	()	_	÷
$x^{1/y}$	7	8	9	×	x^y	7	8	9	×	x^y	7	8	9	×
2^x	4	5	6	-	10^x	4	5	6	-	10^x	4	5	6	-
n!	1	2	3	+	log	1	2	3	+	log	1	2	3	+
e^x	ANS	0	,	=	ln	ANS	0	,	=	ln	ANS	0	,	=

Przełączenie aplikacji w tryb naukowy pozwala użytkownikowi na wykonywanie bardziej skomplikowanych operacji. Za pomocą przycisku „Switch” użytkownik może przełączać się pomiędzy widokiem gotowych szablonów, które znacząco ułatwiają wykonywanie obliczeń. Dodatkowo w tym oknie zostały zaimplementowane takie stałe jak liczba pi czy liczba e, które pozwolą na łatwiejsze wykonywanie obliczeń. Dodatkowo użytkownik może obliczać logarytmy korzystając z szablonu, który umożliwia wyliczanie logarytmów z dowolną podstawą(logbaza_liczba) lub wpisując logliczba uzyska logarytm o domyślnej podstawie 10, ln pozwala na obliczenie logarytmu o podstawie e. Dodatkowo „|x|” pozwala na obliczenie wartości bezwzględnej z wyniku.

Widok po przełączeniu kalkulatora na programistyczny.

Standard		Scientific		Converter	
	HEX	DEC	OCT	BIN	
A	C	DEL	
B	()	CE	÷	
C	7	8	9	×	
D	4	5	6	-	
E	1	2	3	+	
F	Operation	0	ANS	=	

Kalkulator programistyczny pozwala użytkownikowi na wykonywanie obliczeń na innych systemach niż dziesiętny. Po wybraniu systemu w jakim użytkownik będzie wykonywał działania odpowiednie przyciski zostaną odblokowane a w lewym „pustym” przycisku pojawi się aktualnie aktywny system liczbowy. Przycisk „Operation” otwiera dodatkowe okienko za pomocą którego użytkownik może wprowadzać do wyrażenia operatory bitowe.

Widok okienka z operacjami bitowymi.

AND	OR	NOT	XOR
-----	----	-----	-----

Widok po przełączeniu kalkulatora na konwerter.

Standard	Scientific	Programmer
Input:		
<input type="text"/>		
Output:		
<input type="text"/>		
From:		To:
HEX	CONVERT	HEX
DEC		DEC
OCT		OCT
BIN	TO INPUT	BIN

Konwerter pozwala użytkownikowi na przekształcanie liczb pomiędzy wybranymi systemami liczbowymi. Aby konwerter działał poprawnie należy wybrać system z którego będziemy przekształcać liczbę(górny zablokowany przycisk) na system docelowy(dolny zablokowany przycisk). Po kliknięciu przycisku „CONVERT” w dolnym okienku pojawi się przekształcona liczba i za pomocą przycisku „TO INPUT” użytkownik może przekazać ją do okienka wejściowego.

Źródła

<https://blog.helion.pl/mvc-wprowadzenie-wzorca/>

<https://www.geeksforgeeks.org/>

<https://stackoverflow.com/>

<https://pypi.org/project/PyQt5/>

<https://docs.python.org/3/>

[https://pl.wikipedia.org/wiki/Pami%C4%85tka_\(wzorzec_projektowy\)](https://pl.wikipedia.org/wiki/Pami%C4%85tka_(wzorzec_projektowy))

[https://pl.wikipedia.org/wiki/Fasada_\(wzorzec_projektowy\)](https://pl.wikipedia.org/wiki/Fasada_(wzorzec_projektowy))