

Politechnika Śląska  
Wydział Matematyk Stosowanej  
Kierunek Informatyka

Gliwice, 05.02.2020

Programowanie I  
**projekt zaliczeniowy**

***"Debt-free"***

**Kacper Moll gr. lab. 3E**

## 1. Opis projektu.

Program „Debt-free” jest na pozór bardzo prostym narzędziem umożliwiającym momentalne rozwiązanie problemu z rozliczeniami. Wyróżnia się swoją czytelnością oraz prostotą w obsłudze – co jest bardzo ważne dla każdego użytkownika. W projekcie są również dwie dodatkowe funkcje: kalkulator oraz możliwość podglądu poprzednich zrzutek.

## 2. Wymagania

Opis funkcjonalności projektu w postaci listy :

- Menu – okno początkowe, do którego program automatycznie wraca po ukończeniu operacji;
- Nawigacja pomiędzy funkcjami proponowanymi przez program jedynie za pomocą numerów klawiatury;
- Funkcja „nowa zrzutka” będąca główną częścią programu oferuje klarowne poprowadzenie użytkownika do sfinalizowania rozliczeń;
- Funkcja „kalkulator” jest dopełnieniem całego programu w tematyce finansów, dzięki niemu wszystkie obliczenia użytkownik jest w stanie wykonać bez wychodzenia z programu;
- Funkcja „podgląd poprzednich zrzutek” umożliwia odtworzenie wyników rozliczeń każdej poprzedniej zrzutki. Możliwe jest to dzięki zapisie do pliku;
- Funkcja „zmiana motywu” pozwala na dostosowanie wyświetlanej grafiki do potrzeb użytkownika
- W przypadku błędu strumienia wejścia program usuwa błąd, informuje użytkownika o dokonaniu pomyłki i prosi o ponowne wpisanie wartości.

## 3. Przebieg realizacji

Najtrudniejszym, jak również najważniejszym etapem projektu było dla mnie wymyślenie, a później zaimplementowanie w kodzie algorytmu, dzięki któremu program rozlicza uczestników każdej zrzutki.

Zasada działania tego algorytmu:

Pierwszym etapem jest wyznaczenie długu każdego uczestnika zrzutki.

Dokonałem tego poprzez odjęcie kosztu zamówienia od wartości wyłożonych

pieniędzy danego uczestnika. Dzięki tej operacji dostałem informacje kto udzielił kredytu, a kto się zadłużył. Następnie eliminuje osoby z długiem równym zero, ponieważ one w rzeczywistości nie biorą udziału w zrzutce. Kolejnym etapem algorytmu jest posortowanie uczestników na podstawie wartości długu. W wyniku czego uczestnicy są uporządkowani od największego dłużnika, do najbardziej stratnej osoby. Kluczową i ostatnią operacją algorytmu jest porównywanie długów (bezwzględna wartość) pierwszego i ostatniego. Dokonuję transakcji pieniędzy pomiędzy nimi, tak, aby dług przynajmniej jednego z nich zrównał się zero, wyłączam go z dalszych transakcji. Dzięki temu liczba transakcji nie może być wyższa od  $(u-1)$ , gdzie  $u$ -liczba uczestników. Algorytm ten wykonuje się, aż wszyscy będą rozliczeni.

Algorytm ten zaimplementowałem korzystając ze struktur oraz wektorów. Stworzyłem dwie struktury, pierwsza z nich przechowuje dane wprowadzane przez użytkownika, druga przechowuje dane obliczone przez program. Ułatwia to rozumienie kodu. Stworzyłem dwa wektory tych struktur. Dzięki możliwościom tego kontenera możliwe było zrealizowanie wszystkich etapów algorytmu.

Mając już najważniejszą funkcjonalność programu zajęłem się dodatkowymi funkcjami, oraz szatą graficzną.

Cały kod napisałem dzieląc go na liczne funkcje oraz osobne pliki .cpp: main.cpp, definicje\_funkcji.cpp, info\_o\_programie.cpp, kalkulator.cpp, nowa\_zrzutka.cpp, poprzednie\_zrzutki.cpp. Stworzyłem również plik nagłówkowy – deklaracje.h, w celu deklaracji struktur oraz funkcji. Wszystko po to aby łatwiej się odnajdywać w kodzie.

Posłużyłem się biblioteką zewnętrzną <windows.h>, aby skorzystać z funkcji Sleep() oraz system(„Clor \_\_”) dzięki którym uatrakcyjniłem wyświetlanie treści. Kolejna biblioteka <conio.h> pozwoliła mi stworzyć wybranie odpowiedniej opcji, bez konieczności zatwierdzania enterem.

#### 4. Instrukcja użytkownika

##### „Menu główne”

Aby wejść w wymienioną funkcję programu wystarczy wpisać numer na klawiaturze odpowiadający numerowi danej funkcji.

```
## ##### ## ##### ##### ## ##### ##
# # # # # # # # # # #
# # # # # # # $ $ # # # # #
# # # # # # # # # # #
## ##### ## # # # ##### ##

MENU GŁÓWNE
-----
$ 1. Nowa zrzutka $
$ 2. Kalkulator $
$ 3. Podgląd poprzednich zrzutek $
$ 4. Informacje o programie $
$ 5. Wyjdź z programu $
-----

Wybierz: _
```

##### „Nowa zrzutka”

Należy wprowadzić dane, o które prosi nas program, za pomocą klawiatury.

	Powinien zapłacić	Zpłacił
Uczestnik 1	0	0
Uczestnik 2	0	0
Uczestnik 3	0	0

Wprowadź nazwę uczestnika nr 1: \_

Program zwraca poprawne rozliczenie zrzutki pieniężnej oraz pyta czy zapisać zrzutkę w historii. Należy wprowadzić odpowiedni numer z klawiatury. W przypadku zapisu należy wprowadzić nazwę.

	Powinien zapłacić	Zpłacił
Kacper	21	74
Artur	22	0
Kamil	31	0

Kamil powinien oddać 31zł uczestnikowi Kacper.  
Artur powinien oddać 22zł uczestnikowi Kacper.

Zapisać tą zrzutkę?  
1. Zapisz  
2. Nie zapisuj

#### 5. Podsumowanie i wnioski.

Myślę, że mogę śmiało stwierdzić, iż udało mi się zrealizować wszystkie założenia. Pracując nad tym projektem odkryłem w sobie zapal do nauki programowania. Przez długi czas wymyślenie poprawnego algorytmu sprawiało mi trudność, lecz później już wiedziałem, że idę dobrą drogą.

Widzę ogromny potencjał w tym programie, a inicjatywa stworzenia właśnie go była skutkiem wielu frustracji z powodu nierozliczonych długów. Nigdy nie

*korzystałem z podobnej aplikacji, dowiedziałem się o ich istnieniu w trakcie tworzenia Debt-free.*

*Z funkcjonalności, o którą chciałbym wzbogacić mój program jest na pewno obsługa kalkulatora podczas tworzenia nowej zrzutki (w tej wersji należy najpierw posłużyć się kalkulatorem). Kolejna funkcja którą chciałbym dodać to możliwość powrotu w każdym momencie do Menu głównego. W tej chwili myślę, że oprócz tych dwóch funkcji, niczego nie brakuje mojemu programowi, ponieważ nie jestem zwolennikiem ładowania niezliczonych funkcjonalności. Program jest przyjazny jeśli jest łatwy w użyciu.*

*Wiele się nauczyłem dzięki temu projektowi, ale jeszcze więcej wiadomości utrwaliłem. To był mój pierwszy tak duży projekt. W przyszłości zamierzam zacząć naukę pisania programów z użyciem biblioteki SFML, więc może i Debt-free czeka graficzna przemiana 😊*