Politechnika Śląska

Wydział Matematyk Stosowanej

Kierunek Informatyka

Gliwice, 13.01.2020

Programowanie I

**Projekt Zaliczeniowy**

**"*Programy użytkowe oraz Kółko i Krzyżyk*"**

**Radosław Terelak gr. lab. G**

**1. Opis projektu**

*Szukając inspiracji co do tematu projektu, nie natknąłem się na jeden konkretny duży, który zainteresowałby moją osobę. Postanowiłem więc zaimplementować kilka mniejszych pomysłów, które są moim zdaniem całkiem użyteczne. Zbudowałem kalkulator liczb rzymskich, generator haseł oraz kalkulator oszczędności. Zadbałem również o aspekt rozrywkowy implementując grę "Kółko i Krzyżyk" dla jednego, bądź dwóch graczy.*

**2. Funkcjonalności**

***Kalkulator liczb rzymskich:***

*- podstawowe operacje matematyczne,*

*- konwersja wyniku na postać dziesiętną,*

*- sprawdzanie poprawności zapisu podanych przez użytkownika liczb,*

*- wyświetlanie historii operacji.*

***Generator haseł:***

*- wybór długości oraz innych parametrów hasła,*

*- automatyczne kopiowanie wygenerowanego hasła do schowka.*

***Kalkulator oszczędności:***

*- użytkownik podaje obecny kapitał i jaką kwotę chce zgromadzić,*

*- obliczanie oszczędności na podstawie kapitału i oprocentowania podanego przez użytkownika,*

*- generowanie rocznych symulacji oszczędności,*

*- podanie dokładnego roku, w którym cel zostanie osiągnięty,*

*- eksportowanie danych do pliku .txt.*

***Gra "Kółko i Krzyżyk":***

*- gra prowadzona jest przez dwóch graczy lub gracza i bota naprzemiennie,*

*- informacja o zwycięzcy,*

*- możliwość rozegrania rewanżu,*

*- jeśli rozgrywana jest więcej niż jedna partia na ekranie ukazuje się tablica wyników.*

**3. Przebieg realizacji**

Kod projektu w całości znajduje się w pliku ***Projekt\_Terelak.cpp***.

***Kalkulator liczb rzymskich:***

Ta część programu składa się z czterech funkcji.

Pierwsza to ***kalkulator\_start()*** będąca początkiem tej sekcji programu. Służy  
do wprowadzania danych przez użytkownika i wykonywania jego poleceń.

Druga to ***kalkulator\_oblicz()***, która pobiera dane użytkownika i oblicza wynik.

Trzecia i czwarta funkcja odpowiada za zamianę danych wartości na system dziesiętny bądź system rzymski, odpowiednio ***rzym\_na\_dec()*** oraz ***dec\_na\_rzym()***.

Biblioteki przydatne w tej części kodu to ***string*** i ***map***. Pozwoliły one wygodnie manipulować danymi. Były bardzo przydatne przy konwersji wartości  
z rzymskich na dziesiętne i na odwrót.

***Generator haseł:***

Ta część kodu składa się z dwóch funkcji.

Pierwsza to ***generator\_hasel\_start()***, służąca do wyboru typu generowanego hasła oraz późniejszego wypisania go na ekranie.

Druga to ***generator\_hasel()***, służąca do generacji losowego hasła na podstawie parametrów określonych przez użytkownika. Po wygenerowaniu korzystam  
z funkcji pomocniczej ***do\_schowka()*** pozwalającej skopiować wynik do schowka podręcznego użytkownika, dzięki czemu hasło jest gotowe do wklejenia  
w potrzebnym miejscu.

Biblioteki przydatne w tej części kodu to ***random*** oraz ***string***. Dzięki bibliotece ***random*** generuję losową liczbę z określonego przedziału pozwalającą stworzyć unikalny ciąg znaków będący wynikiem.

***Kalkulator oszczędności:***

Ta część kodu składa się jednej funkcji nazwaną ***kalkulator\_oszczednosci()***. Przyjmuje wartości od użytkownika i na ich podstawie prowadzi symulację.

Biblioteki przydatne w tej części kodu to ***WinBase.h*** oraz ***fstream.*** Pierwsza pozwala mi pobrać aktualny rok, aby okres użytkowania programu nie był ograniczony i mimo upływu lat dalej pokazywał poprawną wartość. Druga posłużyła mi do obsługi plików.

***Gra "Kółko i Krzyżyk":***

Ta część kodu składa się z trzech funkcji.

Pierwsza to ***kolko\_krzyzyk\_start()*** służąca do wyboru trybu gry i prowadzenia rozgrywki.

Druga to ***rysuj\_tablice()***, która generuje tablicę gry i aktualizuje jej wartości podczas rozgrywki.

Ostatnia funkcja to ***czy\_wygrana()***, która analizuje obecnie wypełnione pola  
i zwraca wynik rozgrywki.

Biblioteki przydatne w tej części kodu to ponownie ***random*** oraz ***algorithm***. Biblioteka ***random*** pozwala mi generować losowe pola przez bota, a dzięki bibliotece ***algorithm*** korzystam z funkcji ***any\_of()*** pozwalającej mi w łatwy sposób sprawdzać czy wylosowane pole przez bota jest już zajęte na planszy.

Biblioteka ***Windows.h*** została użyta w funkcji pomocniczej ***do\_schowka()***,  
a biblioteka ***iomanip*** pomogła mi lepiej wyświetlić dane w kilku miejscach.

**4. Instrukcja użytkownika**

***Kalkulator liczb rzymskich:***

*Program ten wykonuje podstawowe operacje arytmetyczne na liczbach rzymskich.*

Użytkownik podaje dwie liczby w systemie rzymskim oraz znaku działania. Jeśli podana liczba zawiera niedozwolone znaki, bądź nie istnieje w systemie rzymskim pojawi się informacja o błędnych danych i użytkownik zostanie poproszony o podanie poprawnej wartości.

Jeśli wynik działania przekroczy zakres **<1-3999>** liczba zostanie automatycznie przedstawiona w systemie dziesiętnym, jeśli nie to zostanie przedstawiona  
w systemie rzymskim.

Możliwa jest konwersja wyniku w postaci rzymskiej na dziesiętną, należy wprowadzić **„T”**, bądź **„t”** z klawiatury gdy na ekranie pojawi się adekwatny komunikat.

Użytkownik może wykonać więcej niż jedno działanie, wystarczy podczas komunikatu o kolejnych działaniach wprowadzić **„T”** lub „t” z klawiatury.

Gdy użytkownik zakończy wszystkie przewidziane przez niego obliczenia możliwe jest również wyświetlenie **historii wszystkich operacji** jakie zostały wykonane w danej sesji tego programu. Zostaną one wyświetlone w czytelny sposób uwzględniając dane wejściowe, znak operacji oraz wynik końcowy.  
Aby to zrobić należy podczas komunikatu o historii wprowadzić **„T”** lub **„t”**  
z klawiatury.

***Generator haseł:***

*Program ten generuje losowo hasła o określonym typie i długości.*

Użytkownik wybiera z czego ma się składać jego hasło. Do wyboru są trzy opcje: ***tylko* *litery***, ***litery i cyfry****,* ***litery*, *cyfry i znaki specjalne***. Po tym wyborze użytkownik jest proszony o podanie długości hasła z przedziału **<5-30>**.  
Na ekranie ukaże się wygenerowane hasło o danych parametrach. Jeśli nie było problemu ze skopiowanie do schowka, pojawi się adekwatny komunikat. Hasło będzie dostępne w schowku podręcznym użytkownika i może zostać wklejone za pomocą kombinacji klawiszy **CTRL+V**.

Użytkownik może wygenerować kolejne hasło jeśli sobie tego życzy. Gdy pojawi się adekwatny komunikat należy wprowadzić **„T”**, bądź **„t”** z klawiatury.

Każde kolejne wygenerowane hasło nadpisuje zawartość schowka użytkownika!

***Kalkulator oszczędności:***

*Program pozwala zasymulować w jakim czasie możemy zgromadzić określoną kwotę, przy znajomości naszego kapitału oraz rocznego oprocentowania.*

Użytkownik jest proszony o podanie **kapitału początkowego**, który jest większy od zera oraz nie przekracza miliarda.

Następnie podaje **wartość kwoty**, którą chciałby zgromadzić. Musi być **większa** od podanego wcześniej kapitału!

Na koniec podaje **roczne oprocentowanie** z przedziału **(0-50)**.

Po wprowadzeniu poprawnych danych na ekranie wygeneruje się coroczny wykaz kapitału, który będzie się powiększał o podane oprocentowanie.

Na końcu zostanie wypisana informacja, w jakim konkretnie roku dany cel zostanie osiągnięty.

Jeśli nie było problemów, w folderze z programem zostanie utworzony plik ***„Symulacja\_Oszczednosci.txt”*** zawierający dane przedstawione na ekranie programu. Proces zostanie potwierdzony adekwatnym komunikatem  
na ekranie.

***Gra "Kółko i Krzyżyk":***

*Program umożliwiający zagranie w popularną grę „Kółko i Krzyżyk” w trybie jedno-, bądź dwuosobowym.*

Rozgrywkę zaczynamy od wyboru trybu gry.

Następnie na ekranie generuje się plansza oraz informacja o graczach oraz ich znakach, które będą wypełniać wybrane przez nich pola.

Gracze, bądź gracz wypełniają pola naprzemiennie podając cyfry z zakresu  
<1-9>. Jeśli gracz poda wartość spoza zakresu może wprowadzić ją ponownie. Jeśli poda wartość już wcześniej wykorzystaną będzie to uznane za błąd!

Jeśli gramy w trybie pojedynczego gracza, Bot Marek zna nasze wcześniejsze ruchy i nie popełnia błędów. Jednak wypełnia planszę w sposób losowy.

Gdy, któryś z graczy, bądź bot wygra, na ekranie pojawia się komunikat  
o zwycięzcy i proces podawania cyfr zostaje przerwany.

Możliwy jest do rozegrania rewanż, aby to zrobić należy po adekwatnym komunikacie wpisać „T”, bądź „t” z klawiatury.

Jeśli zdecydujemy się na rewanż nad planszą pojawi się tablica wyników pokazująca obecny stan rozgrywki.

Aby zakończyć grę należy zrezygnować z opcji rewanżu.

**5. Podsumowanie i wnioski**

Wybierając wstępne funkcjonalności postawiłem na rzeczy, które wiedziałem, że jestem w stanie wykonać. Podczas realizacji wykorzystałem kilka nowych pomysłów i lekko ulepszyłem wstępne założenia. Sam proces twórczy przeszedł bardzo dobrze, bez większych komplikacji. Tworząc ten projekt poznałem kilka nowych bibliotek i nauczyłem się korzystać z nowych funkcji oferowanych przez C++. Rozwinąłem swoje umiejętności projektowe począwszy od wstępnych założeń i procesu realizacji, aż po samo pisanie kodu i jego podział na mniejsze części co sprawia, że jest czytelniejszy i prostszy w implementacji.

Myślę, że dość dobrze zabezpieczyłem cały program przed wpisywaniem błędnych danych i próbach wykonania czegoś nieprzewidzianego. Do poprawy są cin.ignore(), ponieważ w niektórych miejscach w określonej sytuacji robi się ich za dużo i trzeba je „przeklikać”. Również funkcja kopiująca dane do schowka nie zawsze działa prawidłowo jeśli schowek wcześniej przetrzymuje już jakieś dane. Może np. skopiować jedynie fragment wygenerowanego ciągu znaków.

Z pewnością projekt ten można dalej rozwinąć, a kod bardziej zoptymalizować. Jeśli miałbym nad czymś popracować to spróbowałbym wyświetlić dane  
w konsoli w czytelniejszy i ładniejszy sposób ale podczas prac nad projektem nie wpadłem na żadne, bardziej optymalne rozwiązanie.