**Kasa zapomogowo-pożyczkowa**

Autorzy:

/hidden/

/hidden/

/hidden/

Spis treści:

# Krótki wstęp

# Szczegółowa analiza wymagań użytkownika

# Podstawowe obiekty i słownik danych globalnych

# Opis działania algorytmów wraz z wyszczególnieniem punktów krytycznych

# Ustalenie co należy przechowywać w bazie danych, ilustracja bazy poprzez tabele

# Projekt interfejsu użytkownika. Założenia ogólne, lista dialogów, ekranów, formatek

# Plan testowania

# Proponowany harmonogram implementacji projektu, podział zadań pomiędzy członków zespołu

1. **Krótki wstęp:**

Projekt utworzony dla Kasy Zapomogowo-Pożyczkowej będzie programem obsługującym system pożyczkowy rzeczonej Kasy. Klient wymaga, aby system Kasa Zapomogowo-Pożyczkowa realizował poniższe wymagania:

1. Możliwość łączności z bazami danych, dotyczących klientów banku i pożyczek danych klientów**.**
2. Interfejs użytkownika na być wykonany klarownie
3. Możliwość dodawania, usuwania, modyfikowania danych dotyczących klientów
4. Możliwość wyszukiwania w liście klientów zadłużenia na określony dzień
5. Sortowanie listy klientów po nazwisku i po wysokości pożyczki
6. Możliwość sprawdzenia, którzy klienci kończą spłacać pożyczkę w tym miesiącu

Wykorzystane zostaną takie narzędzia jak:

Struktura opisująca kolejne osoby z bazy danych.

Imię

Nazwisko

PESEL

Numer Dowodu

Ulica

Nr mieszkania

Kod pocztowy

Miasto

Numer1 (numer identyfikujący dane pierwszej pożyczki)

Numer2

Numer3

Numer4

Numer5

Dług (łączne zadłużenie klienta)

Struktura opisująca poszczególne pożyczki:

Numer umowy o pożyczkę

Kwota Pożyczki

Data Pożyczki

Ilość miesięcy do spłaty

Stan na dzień bieżący (ile teraz osoba jest winna bankowi)

Oprocentowanie

(oraz kilka elementów potrzebnych do obliczeń)

W dokumentacji przedstawione zostaną przedstawione kolejne wymagania użytkowników, słownik globalnych danych (zmienne opisujące dane, nazwy funkcji), opis działania algorytmów, projekt interfejsu użytkownika, planowanie testów oraz plan implementacji projektu wraz z harmonogramem działania członków zespołu.

1. **Szczegółowa analiza wymagań użytkownika:**

Wymagane jest, aby:

1. Interfejs użytkownika zawierał komentarze opisujące poszczególne opcje (np. do opcji wyszukaj szczególny komentarz //wyszukuje osobę o podanym nazwisku spośród bazy danych……
2. System obsługiwał bazę danych zawierającą do 2000 klientów i 10 000 pożyczek.
3. System realizował wymienione w poprzednim punkcie funkcje
4. **Podstawowe obiekty i słownik danych globalnych**

Pliki:

- Dane o pożyczkobiorcach zapisane w pliku osoba.txt

- Dane o pożyczkach w pliku p.txt

Zmienne globalne:

- int b\_rok, b\_miesiac, b\_dzien – zmienne służące do przechowywania bieżącej daty (wypełniane przez użytkownika)

- int i – zmienna pomocnicza

- int LK – zmienna służąca do przechowywania aktualnej liczby klientów

- int LP – zmienna służąca do przechowywania aktualnej liczby pożyczek

Struktury:

- osoba – typ struktury służący do obsługi bazy klientów

- bank – typ struktury służący do obsługi bazy pożyczek

Funkcje:

// główne – zawierające kluczowe elementy programu

void baza\_k(); //wyświetla bazę klientów

void mod\_klienta(); //modyfikacja danych klienta

void dodaj\_klienta(); //dodanie klienta do bazy danych

void usun\_klienta(); //usuniecie klienta z bazy danych

void baza\_p(); //wyświetlanie bazy pożyczek

void sort\_nazw(); //wyświetlanie listy klientów posortowanej po nazwisku

void sort\_kwota(); //wyświetlanie listy klientów posortowanej po wysokości pożyczki

void splata\_teraz(); //wyświetlanie listy klientów którzy skończą spłacać jakąś pożyczkę w

//w tym miesiącu

void zadluzenie(); //sprawdzanie ile zadłużony jest dany klient

void zapisz(); //zapisuje bazę danych po wprowadzonych zmianach

void zapisz\_p(); //zapisuje bazę pożyczek

void menu(); //funkcja do obsługi interfejsu

// pomocnicze – funkcje pomniejsze, najczęściej zawierające się wewnątrz funkcji głównych

void wielka\_litera(); //poprawia na wielka literę, reszta małe

int pesel\_check(); //sprawdza poprawność daty w peselu (nie sprawdza sumy kontrolnej)

int dowod\_check(); //sprawdza poprawność wpisanego nr dowodu

void ulica\_correct(); //poprawia format wpisanej ulicy/miasta

int kod\_check(); //sprawdza poprawność wpisanego kodu pocztowego

int alfabet(); //sprawdza który z podanych stringów jest bardziej alfabetyczny

void oblicz(); //obliczenia związane z bazą pożyczek

void data(); //pobieranie aktualnej daty od użytkownika

int znajdz(); //znajduje klienta o danym numerze pożyczki

void delay(); // „upiększanie”

# Opis działania algorytmów wraz z wyszczególnieniem punktów krytycznych

**Dodawanie klienta:**

Dopisanie danych klienta na końcu bazy klientów

Zwiększenie LK o 1

Wygenerowanie nowych numerów pożyczek (pięciu)

Zwiększenie LP o 5

Obowiązkowe przyznanie przynajmniej jednej pożyczki

**Usuwanie klienta:**

Odszukanie danego klienta w bazie

Usunięcie jego danych

Przesunięcie i przenumerowanie pozostałych klientów

Zmniejszenie LK o 1

Wyzerowanie pożyczek o numerach

**Modyfikowanie danych klienta:**

Odszukanie danego klienta w bazie

Zastąpienie jego starych danych nowymi, wprowadzanymi z klawiatury przez użytkownika

Częściowe sprawdzanie spójności podawanych danych oraz poprawa formatu dla późniejszego poprawnego wczytywania z pliku

**Sortowanie listy klientów po nazwisku:**

Sortowanie bąbelkowe połączone z funkcją sprawdzającą który z podanych stringów jest bardziej alfabetyczny

**Sortowanie listy klientów po wysokości pożyczki:**

Utworzenie dodatkowej struktury typu bank

Sortowanie (bąbelkowe) nowej struktury pod kątem wysokości pożyczek

Wypisanie kwot pożyczek w kolejności malejącej wraz z przypisanymi do nich klientami

**Sprawdzanie zadłużenia dla osoby o danym numerze:**

Obliczenie i wyświetlenie parametru „dlug” według podanej na początku działania programu daty bieżącej

**Możliwość sprawdzenia, którzy klienci kończą spłacać pożyczkę w tym miesiącu:**

Wyświetlenie klientów dla których przynajmniej jedna pożyczka zostanie spłacona w przeciągu następnego miesiąca

# Ustalenie co należy przechowywać w bazie danych, ilustracja bazy poprzez tabele

**Struktura osoba:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Imię | Nazwisko | Pesel | Nr dowodu | Ulica | Nr  Mieszkania | Kod  Pocztowy | Miasto | Numer pożyczki  (1-5) | Dług |
| int | string | string | string | string | string | string | string | string | int | float |

**Struktura bank:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Numer | Kwota | Data | Na ile  m. | Procent | Ile m.  już spłac. | Ile m.  pozostało  do spłac. | Jaka  kwota  pozostała  do spłac. | Rata  kapitałowa | Odsetki | Rata  łączna |
| int | int | int | int | float | int | int | float | float | float | float |

1. **Projekt interfejsu użytkownika:**

Funkcja data() pobiera bieżącą datę od użytkownika

Funkcja menu() wypisuje poniższy interfejs:

1. Wyświetl bazę klientów
2. Modyfikuj dane klienta
3. Dodaj klienta
4. Usuń klienta
5. Wyświetl bazę pożyczek
6. Wyświetl listę klientów posortowaną po nazwisku
7. Wyświetl listę klientów posortowaną po wysokości pożyczki
8. Sprawdź, którzy spośród klientów kończą spłacać pożyczkę w tym miesiącu
9. Sprawdź zadłużenie danego klienta
10. Zapisz
11. Zapisz i zakończ
12. Zakończ

Podaj numer:

1. **Plan testowania**

Testy w trakcie pisania kodu:

- Testy działania konkretnych funkcji np. sortowania po nazwisku, w trakcie implementacji i natychmiastowe poprawki ewentualnych błędów działania.

- Próbne wypisywanie tablicy struktur do pliku dla sprawdzenia spójności wprowadzonych/wyjściowych danych.

Testy po zaimplementowaniu interfejsu użytkownika:

- Sprawdzenie wszystkich opcji menu, poprawa ewentualnych błędów działania.

Działania dodatkowe (opcjonalne) :

- Korekta wyświetlania danych dla łatwiejszego użycia przez użytkownika

# Proponowany harmonogram implementacji projektu, podział zadań pomiędzy członków zespołu

|  |  |  |
| --- | --- | --- |
| /hidden/ | /hidden/ | /hidden/ |
| - dokumentacja  - interfejs  - testowanie  - spójność | -dokumentacja  - obsługa plików  - in/out danych  - funkcje/algorytmy | - dokumentacja  - sortowanie  - działania na bazie danych  - funkcje/algorytmy |