

# Książka adresowa - koncept

## Funkcjonalności:

1. Dodanie danych kontaktowych.
2. Wypisanie kontaktu o zadanych parametrach.
3. Zapis i wczytywanie kontaktów przy zamknięciu/otwarcu programu.
4. Przeglądanie listy kontaktów.
5. Usuwanie i edytowanie kontaktów.
6. Wysłanie do adresata maila o treści “mam twojego maila”.

## Proponowana realizacja:

Program zostanie zaimplementowany z tekstowym interfejsem.

Po otwarciu programu, dane kontaktowe zostają wczytane z pliku tekstowego. Na ich podstawie zostaną utworzone obiekty klasy “Contact”. Obiekty zostaną zapisane w tablicy “contacts”.

Następnie zostanie wywołana funkcja “scrollContacts()” do której zostanie przekazana tablica kontaktów. Będzie ona odpowiedzialna za wypisywanie i przeglądanie kontaktów oraz wywoływanie kolejnych funkcji (jeśli nie ma żadnych kontaktów w książce adresowej wyświetlany jest stosowny komunikat i opcja dodania kontaktu). Po wprowadzeniu odpowiedniego znaku z klawiatury w funkcji będą wykonywane następne czynności:

- Przejście do następnego/poprzedniego kontaktu i wypisanie go w terminalu
- Usuń - wywołanie funkcji deleteContact() usuwającej kontakt po czym wyświetlenie pierwszego kontaktu z uaktualnionej listy bądź wyświetlenie komunikatu o braku kontaktów
- Wyślij powiadomienie - wywołanie funkcji sendNotification() wysyłającej powiadomienie na adres email danego kontaktu

- Edytuj kontakt - wywołanie funkcji `editContact()` pozwalającej na edycję wyświetlanego kontaktu. Po wywołaniu funkcji nadal jest wyświetlany ten sam kontakt (nie następuje przejście do pierwszego kontaktu z listy tak jak w przypadku usuwania).
- Dodaj kontakt - wywołanie funkcji `addContact()` dodającej nowy kontakt. Przejście do ostatniego kontaktu.
- Wyszukaj - wywołanie funkcji `findContact()` szukającej kontakt do osoby o podanym imieniu, nazwisku i adresie email. Gdy taki kontakt zostanie znaleziony funkcja `scrollContacts()` przechodzi do znalezionej elementu. W innym wypadku wyświetlany jest komunikat o braku takiej osoby i wyświetlony zostaje pierwszy kontakt z listy.
- Zamknij program - wywołanie funkcji `closeProgram()` zapisującej wszystkie kontakty do pliku tekstowego po czym zakończenie funkcji `scrollContacts()` skutkujące zakończeniem wykonywania całego programu.

## Specyfikacja funkcji:

`loadContacts()` - funkcja ładuje kontakty z pliku tekstowego. Jeśli taki nie istnieje, to go tworzy. Funkcja zwraca listę kontaktów.

`deleteContact(array, id)` - funkcja otrzymuje tablicę kontaktów oraz indeks obiektu do usunięcia. Funkcja zwraca zaktualizowaną tablicę.

`sendNotification(email)` - funkcja otrzymuje adres email, na który ma zostać wysłane powiadomienie po czym wysyła je.

`editContact(array, id)` - funkcja otrzymuje tablicę kontaktów oraz indeks kontaktu do edycji. Użytkownik wprowadza po kolei imię, nazwisko, email. Następuje sprawdzenie czy w imieniu, nazwisku, adresie email nie ma spacji. Jeśli warunek jest spełniony dane zostają zaktualizowane i funkcja zwraca zaktualizowaną tablicę. W innym wypadku wyświetlany jest komunikat o błędnych danych i funkcja zwraca tą samą tablicę, którą otrzymała.

`addContact(array)` - funkcja otrzymuje tablicę kontaktów. Użytkownik wprowadza po kolei imię, nazwisko, email. Następuje sprawdzenie czy w

imieniu, nazwisku, adresie email nie ma spacji. Jeśli warunek jest spełniony tworzony jest nowy obiekt klasy "Contact" o podanych parametrach po czym zostaje dodany na końcu tablicy i funkcja zwraca zaktualizowaną tablicę. W innym wypadku wyświetlany jest komunikat o błędnych danych i funkcja zwraca tą samą tablicę, którą otrzymała.

findContact(name, surname, email, array) - funkcja otrzymuje imię, nazwisko adres kontaktu do znalezienia oraz tablicę kontaktów. W przypadku znalezienia podanego kontaktu funkcja zwraca indeks danego kontaktu. W innym wypadku zwraca wartość -1.

closeProgram(array) - funkcja zapisuje kontakty z listy do pliku tekstowego w następujący sposób: imię nazwisko email. Następny kontakt jest zapisywany w następnej linii.