

# Dokumentacja - Projekt - Temat: Obsługa sklepu

## Treść:

Program do obsługi sklepu umożliwiający rejestrację towarów i klientów oraz zamówień. Każdy towar jest opisany atrybutami: identyfikator, nazwa, kategoria, cena. Dane klienta to: imię, nazwisko, adres, email, telefon kontaktowy. Program powinien umożliwiać dodawanie, usuwanie, edycję, wypisywanie, wyszukiwanie towarów oraz klientów. Kupno towarów odbywa się poprzez złożenie zamówienia. Każde zamówienie ma identyfikator, datę złożenia, dane klienta, adres dostawy, dane towaru, liczbę sztuk, cenę, oraz status (w realizacji – stan początkowy po złożeniu, wysłane, anulowane). Program powinien umożliwiać zmianę stanu zamówienia z „w realizacji” na „wysłane” lub na „anulowane”. Powinna być dostępna opcja wyświetlania zamówień złożonych przez danego klienta oraz zamówień złożonych na dany towar. Przy wyświetlaniu list z danymi należy dodać sortowanie wg wybranych pól.

## Przechowywanie danych:

Dane przechowywane są w 3 plikach tekstowych: klienci.txt, produkty.txt, zamowienia.txt.

Plik klienci.txt zawiera informacje w następującej postaci – w jednym wierszy zapisane są wszystkie atrybuty klienta oddzielone spacją w podanej kolejności: imię, nazwisko, adres, email, numer telefonu. Przykład:

Szymon Radziszewski Białystok/kujawska/34 szymonradziszewski@wp.pl 683534234

Plik produkty.txt zawiera informacje w następującej postaci – w jednym wierszy zapisane są wszystkie atrybuty produktu oddzielone spacją w podanej kolejności: id, nazwa produktu, kategoria, cena.

Przykład:

1 Odkurzacz sprzety\_elektroniczne 229.99

Plik zamowienia.txt zawiera informacje w następującej postaci – w jednym wierszy zapisane są wszystkie atrybuty zamówienia oddzielone spacją w podanej kolejności: id zamówienia, id produktu, ilość zamówionych produktów, imię klienta, nazwisko klienta, adres klienta, email, numer telefonu, nazwa produktu, kategoria produktu, adres dostawy, data wysłania, status zamówienia, cena produktu, cena całego zamówienia.

Przykład:

99 2 2 Szymon Radziszewski Białystok/kujawska/34 szymonradziszewski@wp.pl 683534234 Mop sprzety Białystok/Zachodnia/16/25 21/02/2021 in\_progress 100.00 200.00

## Dane w programie:

Dane w programie przechowywane są w postaci list jednokierunkowych. Pierwszy element każdej listy jest ustawiony na NULL – nie ma przydzielonej żadnej pamięci, dodawana jest w funkcjach.

## Przechowywanie programu:

Program przechowywany jest w jednym pliku: main.c.

Zawiera on funkcje do operowania na listach, struktury list, biblioteki, definicje stałych.  
Przykład struktury:

```
typedef struct list_of_product
{
    int id;
    float price;
    char name[N], category[N];
    struct list_of_product * next;
} list_of_product;
```

Plik main.c zawiera: menu programu, wywołania funkcji, zbudowane przeze mnie menu, deklaracje pierwszego elementu list.

Przykład wywołania list:

```
list_of_client* head1 = NULL;
list_of_product* head2 = NULL;
list_of_orders* head3 = NULL;
```

## Funkcje:

Funkcje do operowania na liscie list\_of\_client

```
void list_add_client(list_of_client **head_of_client, char *name, char *last_name, char *adress, char *email, char *phone);
void load_client(const char *fn2, list_of_client **head1);
void list_of_clients_print (list_of_client *head_of_client);
void deallocate_clients (list_of_client **head_of_client);
void remove_client (list_of_client **head_of_client, int number);
void search_client (list_of_client* head_of_client, char *name, char *last_name);
void edit_client (list_of_client **head_of_client, int number, char *name, char *last_name, char *adress, char *email, char *phone);
void save_list_of_client_to_file (list_of_client* head_of_client, const char *fn2);
```

Funkcje do operowania na liscie list\_of\_product

```
void list_add_product (list_of_product **head_of_product, int id, char *name, char *category, float price);
void load_product(const char *fn1, list_of_product **head_of_product);
void list_of_products_print (list_of_product *head_of_product);
void deallocate_product (list_of_product **head_of_product);
void remove_product (list_of_product **head_of_product, int number);
void search_product (list_of_product* head_of_product, char *name);
void edit_product (list_of_product **head_of_product, int number, int id, char *name, char *category, float price);
void save_list_of_product_to_file (list_of_product *head_of_product, const char *fn1);
void remove_order (list_of_orders **head_of_order, int number);
int check_if_id_of_product_appears(list_of_product* head_of_product, int id);
```

Funkcje do operowania na liście list\_of\_orders

```
void save_list_of_order_to_file (const char *fn3, list_of_orders* head_of_order);
void remove_order (list_of_orders **head_of_order, int number);
void insert_order (list_of_orders **head_of_orders, list_of_client* head_of_client, list_of_product*
head_of_product, int number_of_client, int number_of_product, int order_id, int the_number_of_items,
char *date_of_submission, char *delivery_adress);
void list_add_order (list_of_orders **head_of_orders, int order_id, int product_id, int
the_number_of_items, char *name_of_client, char *last_name, char *adress_of_client, char *email,
char *phone, char *name_of_product, char *category, char *delivery_adress, char
*date_of_submission, char *status, float price_of_product, float total_price);
void load_order(const char *fn3, list_of_orders **head_of_orders);
void list_of_order_print (list_of_orders* head_of_order);
void search_order (list_of_orders* head_of_orders, int id);
int check_if_id_of_order_appears (list_of_orders* head_of_orders, int id);
void change_status (list_of_orders* head_of_orders, int id, char *status);
void print_orders_with_same_client (list_of_orders* head_of_order, char *name, char *surname);
void print_orders_with_same_product (list_of_orders* head_of_order, int product_id);
void deallocate_orders (list_of_orders **head_of_orders);
```

Funkcja pomocnicza

```
void Break()
```

**Twórca:**

Imię i nazwisko: Szymon Fiedosiuk

Nr albumu: 109557

Wydział Informatyki PB