

Projekt z Podstaw Programowania

Tworzenie projektu zacząłem od przygotowania plików potrzebnych do realizacji zadania. W tym celu napisałem generator w języku Python, który losuje wartości i tworzy listy produktów w sklepach. Na podstawie wygenerowanych list produktów tworzone są również listy zakupów.

Kod źródłowy Python

```
import random

nazwy = ["chleb", "bulka", "maslo", "mleko", "serek_wiejski", "twarog", "parowki",
"kielbasa", "pomidory", "ogorek",
        "cebula", "papryka", "kapusta", "salata", "woda", "CocaCola", "maka",
"cukier", "sok_pomaranczowy", "banany",
        "jablko", "czekolada", "platk_owsiane", "kefir", "jogurt", "wedlina",
"szynka", "paprykarz"]

id_lista = []

def Generuj_id():
    id = []
    for i in range(10):
        id.append(random.randint(0,9))
    idstr = ''.join(map(str, id))
    return idstr

def Generuj_produkt():
    nazwa = random.choice(nazwy)
    cena = round(random.uniform(1.1, 10.99), 2)
    liczba_sztuk = random.randint(10, 100)
    id = Generuj_id()
    if id not in id_lista:
        id_lista.append(id)
        return id+" "+str(nazwa)+" "+str(cena)+" "+str(liczba_sztuk)
    else:
        while id not in id_lista:
            id = Generuj_id()
            if id not in id_lista:
                id_lista.append(id)
                return id+" "+str(nazwa)+" "+str(cena)+" "+str(liczba_sztuk)

def Generuj_plik(rozmiar, nazwa_sklepu):
    plik = open(nazwa_sklepu, "w")
    for i in range(rozmiar):
        plik.write(str(i+1) + " "+Generuj_produkt() + "\n")

def Generuj_zakupy(rozmiar, nazwa_zakupow):
    zakupy = []
    for i in range(rozmiar):
```

```
        zakupy.append(id_lista[random.randint(0,(len(id_lista)-1))])
plik = open(nazwa_zakupow, "w")
for i in range(rozmiar):
    plik.write(str(zakupy[i])+"\n")

Generuj_plik(100, "sklep1.txt")
Generuj_zakupy(10, "zakupy1.txt")
id_lista.clear()
Generuj_plik(100, "sklep2.txt")
```

Właściwy kod

Pierwszą częścią było stworzenie struktury, która opisuje Produkt, a potem funkcji, która odczyta zasoby sklepu z pliku, oraz wpisze je do listy.

Kolejnym krokiem było stworzenie funkcji do wyświetlania listy, aby zobaczyć, czy wszystko działa poprawnie. Następnym krokiem było napisanie funkcji, która jako parametr dostaje listę i zwraca średnią cenę z wszystkich produktów w sklepie.

Następną rzeczą do zrobienia było stworzenie nowej listy na podstawie parametrów takich jak cena produktu oraz ilość sztuk w magazynie sklepu. Funkcja jako parametry otrzymuje listę, minimalną ilość produktów i maksymalną cenę. Funkcja nie różni się mocno od funkcji wczytującej dane z pliku do listy, więc jej napisanie nie było trudne. Jeżeli produkt spełnia warunki, to jest dodawany do nowej listy.

Do realizacji ostatniej części zadania stworzyłem funkcję, która wczytała do tablicy charów identyfikatory produktów z pliku zakupy.txt. Funkcja jako parametry dostaje nazwę pliku z którego ma odczytać listę zakupów, oraz tablicę, do której ma wczytać zawartość pliku.

Kiedy miałem już tablicę z identyfikatorami z listy z zakupami napisałem funkcję, która policzy cenę za zakupy z listy. W tym celu w pętli while przechodzę po każdym elemencie listy i dla każdego elementu sprawdzam w pętli for czy jakieś id produktu z listy produktów w sklepie jest takie samo jak id w tablicy z zakupami. W tym miejscu pojawił się największy problem, a polegał on na tym, że porównywałem stringi w taki sposób:

```
if(zakupy[i] == pom->identyfikator_produkту)
```

Niestety w C nie można tak porównywać stringów, do czego dojdzie zajęło mi kilka godzin i ostatecznie do porównania id produktów użyłem funkcji:

```
strcmp()
```

Funkcja już działała prawidłowo, więc wszystkie funkcje do realizacji zadania były gotowe. W funkcji main() napisałem warunki do porównania cen ze sklepów i do wybrania "lepszego" sklepu i do użycia go do policzenia ceny za zakupy oraz stworzenia listy z konkretnymi produktami.

Kod źródłowy C

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define rozmiar_ciagu 60
#define zakupy_rozmiar 10
#define ilosc_sztuk 70
#define max_cena 4
```

```
typedef struct Produkt_sklep {
    int lp;
    char identyfikator_produkту[rozmiar_ciagu+1];
    char nazwa[rozmiar_ciagu+1];
    float cena;
    int liczba_sztuk;
    struct Produkt *nast;
} Element_listy;

typedef Element_listy* Produkt;

Produkt Odczyt_z_pliku(const char *nazwa){
    int i;

    FILE* plik = fopen(nazwa, "r");
    if (plik == NULL)
    {
        printf("Bład otwarcia pliku '%s'\n", nazwa);
        return NULL;
    }

    Produkt pierwszy_produkт = NULL, ostatni_produkт;

    int lp1;
    char identyfikator_produkту1[rozmiar_ciagu+1];
    char nazwa1[rozmiar_ciagu+1];
    float cena1;
    int liczba_sztuk1;

    while (fscanf(plik, "%d%s%s%f%d", &lp1, identyfikator_produkту1, nazwa1,
&cena1, &liczba_sztuk1) == 5) {
        Produkt nowy = (Produkt) malloc(sizeof(Element_listy));
        nowy->lp = lp1;
        strcpy(nowy->identyfikator_produkту, identyfikator_produkту1);
        strcpy(nowy->nazwa, nazwa1);
        nowy->cena = cena1;
        nowy->liczba_sztuk = liczba_sztuk1;
        nowy->nast = NULL;

        if (pierwszy_produkт == NULL)
        {
            pierwszy_produkт = nowy;
            ostatni_produkт= nowy;
        }
        else
        {
            ostatni_produkт->nast = nowy;
            ostatni_produkт = nowy;
        }
    }
    fclose(plik);

    return pierwszy_produkт;
}
```

```
void czytaj_zakupy_z_pliku(const char *nazwa, char zakupy[][rozmiar_ciagu]) {
    int i = 0, dlugosc_pliku;
    FILE *plik = fopen(nazwa, "r");

    while (fgets(zakupy[i], rozmiar_ciagu, plik)) {
        zakupy[i][strlen(zakupy[i])-1] = '\0';
        i++;
    }
    fclose(plik);
}

void wypisz_liste(Produkt pierwszy) {
    Produkt pom = pierwszy;
    while (pom != NULL) {
        printf("%d ", pom->lp);
        printf("%s ", pom->identyfikator_produkту);
        printf("%s ", pom->nazwa);
        printf("%f ", pom->cena);
        printf("%d\n", pom->liczba_sztuk);
        pom = pom->nast;
    }
}

float policz_srednia_cene(Produkt pierwszy) {
    float srednia, suma = 0;
    int licznik = 0;
    Produkt pom = pierwszy;
    while (pom != NULL) {
        suma += pom->cena;
        licznik++;
        pom = pom->nast;
    }
    srednia = suma/licznik;
    return srednia;
}

Produkt tworz_liste_progi(Produkt pierwszy, int prog_ilosc, float prog_cena) {
    if (pierwszy == NULL)
        return 1;

    Produkt pierwszy_produkт = NULL, ostatni_produkт;
    Produkt pom = pierwszy;

    while (pom != NULL) {
        if ((pom->liczba_sztuk > prog_ilosc) && (pom->cena < prog_cena))
        {
            Produkt nowy = (Produkt) malloc(sizeof(Element_listy));
            nowy->lp = pom->lp;
            strcpy(nowy->identyfikator_produkту, pom->identyfikator_produkту);
            strcpy(nowy->nazwa, pom->nazwa);
            nowy->cena = pom->cena;
        }
    }
}
```

```
        nowy->liczba_sztuk = pom->liczba_sztuk;
        nowy->nast = NULL;

        if (pierwszy_produkt == NULL)
        {
            pierwszy_produkt = nowy;
            ostatni_produkt = nowy;
        }
        else
        {
            ostatni_produkt->nast = nowy;
            ostatni_produkt = nowy;
        }
    }
    pom = pom->nast;
}
return pierwszy_produkt;
}

float policz_cene_za_zakupy(const char *nazwa, Produkt pierwszy, char zakupy[]
[rozmiar_ciagu]) {

    float cena_za_zakupy = 0;
    int i = 0;

    Produkt pom = pierwszy;

    while (pom!=NULL) {
        for (i=0;i<zakupy_rozmiar;i++) {
            if (strcmp(zakupy[i], pom->identyfikator_produktu)==0) {
                cena_za_zakupy += pom->cena;
            }
        }
        pom = pom->nast;
    }

    return cena_za_zakupy;
}

int main () {

    char zakupy[zakupy_rozmiar][rozmiar_ciagu];
    Produkt sklep1 = Odczyt_z_pliku("sklep1.txt");
    Produkt sklep2 = Odczyt_z_pliku("sklep2.txt");

    float srednia_sklep_1 = policz_srednia_cene(sklep1);
    float srednia_sklep_2 = policz_srednia_cene(sklep2);

    printf("Srednia w sklepie nr 1 = %.2f zl\n", srednia_sklep_1);
    printf("Srednia w sklepie nr 2 = %.2f zl\n", srednia_sklep_2);
    printf("\n");

    if (srednia_sklep_1<srednia_sklep_2) {
        printf("Srednia cena za produkty jest nizsza w sklepie nr 1\n");
    }
```

```
        printf("\n");
        Produkt nowa_lista = tworz_liste_progi(sklep1, ilosc_sztuk, max_cena);
        printf("Lista produktow ktorych cena jest ponizej %d zl i ilosc na stanie
jest wieksza od %d sztuk:\n", max_cena, ilosc_sztuk);
        wypisz_liste(nowa_lista);
        printf("\n");
        czytaj_zakupy_z_pliku("zakupy1.txt", zakupy);
        float cena = policz_cene_za_zakupy("zakupy1.txt", sklep1, zakupy);
        printf("Cena za zakupy z listy to %.2f zl", cena);

    }
    else {
        if (srednia_sklep_2 < srednia_sklep_1) {
            printf("Srednia cena za produkty jest nizsza w sklepie nr 2\n");
            printf("\n");
            Produkt nowa_lista = tworz_liste_progi(sklep2, ilosc_sztuk, max_cena);
            printf("Lista produktow ktorych cena jest ponizej %d zl i ilosc na
stanie jest wieksza od %d sztuk:\n", max_cena, ilosc_sztuk);
            wypisz_liste(nowa_lista);
            printf("\n");
            czytaj_zakupy_z_pliku("zakupy2.txt", zakupy);
            float cena = policz_cene_za_zakupy("zakupy2.txt", sklep2, zakupy);
            printf("Cena za zakupy z listy to %.2f zl", cena);
        }
    }

    return 0;
}
```