

Politechnika Śląska
Wydział Matematyk Stosowanej
Kierunek Informatyka

Gliwice, 01.10.2020

Programowanie I
projekt zaliczeniowy

"Kasa Fiskalna"

Bartłomiej Repeć gr. lab. 2

1. Opis projektu.

Projekt dotyczy kasy fiskalnej, przechowującej dane na temat posiadanych produktów w sklepie, oraz pozwalającej na sprzedaż produktów, a także na wprowadzenie nowych pozycji.

2. Wymagania

- Przechowywanie informacji o produktach dostępnych w sklepie
- Możliwość sprzedaży produktów
- Obliczanie sumy cen zakupionych produktów
- Zapisanie paragonu w pliku tekstowym
- Płatność gotówką – obliczanie reszty koniecznej do wydania
- Możliwość sprawdzenia stanu magazynu sklepu
- Możliwość wprowadzenia dostawy – zwiększenia w systemie ilości posiadanych produktów
- Możliwość wprowadzenia nowych produktów do systemu
- Przezroczystość oraz prostota w użytkowaniu
- Kontrola wartości wprowadzanych przez użytkownika
- Obsługa polskich znaków

3. Przebieg realizacji

Zacząłem tworzenie od zbudowania szkieletu – stworzeniu plików, prostego menu głównego oraz stworzeniu struktury. Następnie utworzenie poszczególnych funkcji – stan magazynu sklepu, sprzedaż, dostawa, oraz nowe produkty. Dostęp do poszczególnych elementów struktury jest realizowany za pomocą tablicy – indeks tablicy obiektu struktury odpowiada jego kodowi w systemie. Wstępnie tablica posiada 100 indeksów, w tym 10 z gotowymi produktami. Pozostałe elementy tablicy są wypełnione „placeholderami”. Zabieg ten służy ułatwieniu wprowadzania nowych produktów. Projekt składa się z 5 plików, a są to:

- Źródło.cpp – główny plik, zawiera proste menu główne oraz odsyła do poszczególnych funkcji
- magazyn.h – plik zawierający dane struktury Produkt przechowującej informacje o produktach w sklepie
- stan.h – plik zawierający funkcję sprawdzającą stan magazynu

- sprzedaz.h – plik zawierający funkcję realizującą sprzedaż oraz zapis paragonu
- dostawa.h – plik zawierający dwie funkcje – zwiększenie ilości produktów w sklepie oraz dodanie nowego produktu do systemu, a także możliwość modyfikacji już istniejących danych o produktach.

4. Instrukcja użytkownika

Po uruchomieniu programu powita nas menu główne:

```
witaj sprzedawco!  
Z której opcji chcesz skorzystać?  
  
1. Tryb sprzedaży  
2. Tryb dostawy  
3. Stan magazynu  
4. Dodaj produkt  
5. Zakończ
```

Wybierając odpowiednią opcję przejdziemy do konkretnej funkcji, którą chcemy wykonać

1. Tryb sprzedaży:

```
Czy chcesz zapisać rachunek? [1 - tak / 0 - nie]
```

Na początku dostajemy wybór – zapisać paragon w pliku, czy nie. Na potrzeby instrukcji użyję wersji z zapisywaniem (bez zapisania działa na identycznych zasadach). Następnie wpisujemy kilka przykładowych

kodów z zakresu 0 do 10 oraz wprowadzamy ilości, tak jak na zrzucie:

```
Czy chcesz zapisać rachunek? [1 - tak / 0 - nie]
1
Podaj kod [-1 spowoduje zakończenie sprzedaży]
2
Wprowadź ilość: 4
4 x Czekolada Milka 23.96 zł
7
Wprowadź ilość: 4
4 x Cukierki Wawel paczka 35.96 zł
9
Wprowadź ilość: 14
14 x Bułka kajzerka 12.46 zł
```

Aby zakończyć sprzedaż wpisujemy -1 jako nasz kod. Następnie zobaczymy taki rezultat:

```
1
Do zapłaty: 72.38 zł.
Wybierz rodzaj płatności: 0 - karta, 1 - gotówka
```

Zsumowana cena naszych zakupów, a także wybór metody płatności. Wybranie płatności kartą wyświetli informację, żeby postępować zgodnie z instrukcjami na terminalu i wprowadzić tam cenę do zapłaty. Natomiast kiedy wybierzemy płatność gotówką, musimy wprowadzić otrzymaną kwotę, a następnie program napisze nam, jaką resztę należy wydać klientowi.

```
Płatność gotówką
Wprowadź otrzymaną kwotę: 100

Reszta: 27.62 zł
Press any key to continue . . .
```

2.Tryb dostawy:

Daje możliwość zwiększenia ilości posiadanego produktu w sklepie

```

Podaj kod produktu (-1 oznacza wyjście z funkcji): 10
Otrzymano: Pomidorki koktajlowe, wprowadź ilość otrzymanego produktu 23
Podaj kod produktu (-1 oznacza wyjście z funkcji): 8
Otrzymano: Makaron Lubella paczka, wprowadź ilość otrzymanego produktu 89
Podaj kod produktu (-1 oznacza wyjście z funkcji):

```

3. Stan magazynu:

Pozwala sprawdzić, ile kosztuje oraz ile jest danego produktu na magazynie, oraz czy produkt o podanym kodzie znajduje się w systemie:

```

Podaj kod produktu, który chcesz sprawdzić [-1 spowoduje wyjście do menu głównego]
8
Nazwa: Makaron Lubella paczka
Stan na magazynie: 118
Cena: 4.59
1
Nazwa: Chipsy Lays
Stan na magazynie: 49
Cena: 4.99
23
Nazwa: Brak produktu
Stan na magazynie: 0
Cena: 0

```

Warto zauważyć, że po wprowadzeniu dostawy, w systemie znajduje się 118 paczek makaronu (przy uruchomieniu programu bazowo jest ich 29)

```
{8, "Makaron Lubella paczka", 29, 4.59}
```

Produkt o kodzie 23 nie istnieje w systemie, więc otrzymaliśmy komunikat o jego braku.

4. Dodaj produkt:

Opcja pozwala nam na dodanie nowego produktu o wybranym kodzie do systemu, lub modyfikację istniejącego już produktu (np. zmiana nazwy, lub ceny).

```

Wprowadź kod pod którym będzie się znajdował Twój produkt (-1 oznacza wyjście z funkcji):
13
Podaj nazwę produktu: Blok Techniczny A4
Podaj cenę: 2.99
Podaj ilość: 26

```

Na początku podajemy kod, później cenę, jaką będzie posiadał produkt, a na końcu jego ilość wprowadzoną do sklepu. Następnie za pomocą opcji

stan magazynu możemy sprawdzić, czy produkt został wprowadzony poprawnie:

```
Podaj kod produktu, który chcesz sprawdzić [-1 spowoduje wyjście do menu głównego]
13
Nazwa: Blok Techniczny A4
Stan na magazynie: 26
Cena: 2.99
```

5. Podsumowanie i wnioski.

Udało się zrealizować każde założenie projektu, jednak realizacja założenia dotyczącego dodawania produktów wymagała zapisania pustych elementów tablicy tzw. „placeholderami”, co nie jest najbardziej optymalnym rozwiązaniem.

W początkowych etapach projektu największą trudność sprawiało dostanie się do konkretnego obiektu naszej struktury za pomocą kodu, dopiero po czasie uznałem, że dobrym rozwiązaniem może być wykorzystanie do tego tablicy, oraz używanie kodów produktów, jako jej indeksów. Problemem również jest możliwość wprowadzenia przez użytkownika niepasujących danych – na każdym etapie wymagana jest kontrola. Ostatnim problemem z jakim stoczyłem walkę było płynne przechodzenie pomiędzy ekranami funkcji, a menu głównym - często pojawiało się miganie i nieskończone zapętlenie jednego etapu.

Możliwości rozwoju projektu:

- Zastąpienie struktur bazami danych umożliwiłoby płynniejszą obsługę większych ilości produktów
- Wprowadzenie kont dla pracowników, a także wprowadzenie sesji
- Generowanie raportów dotyczących sprzedaży oraz podsumowań dziennych
- Monitorowanie stanu gotówki, a także przychodów

- Podzielenie produktów na kategorie, a także rozróżnienie tych sprzedawanych na sztuki, oraz tych sprzedawanych na wagę
- Stworzenie przystępniejszego interfejsu graficznego
- Wprowadzenie odpowiednich podatków VAT na produkty