|  |  |
| --- | --- |
|  | **AKADEMIA GÓRNICZO-HUTNICZA** |

Dokumentacja do projektu

**Vending Machine**

z przedmiotu

**Języki programowania obiektowego**

Elektronika i Telekomunikacja PL, III rok

*Wojciech Przybyło*

Poniedziałek 19:25

prowadzący: Rafał Frączek

21.01.2020r.

# Opis projektu

Projekt opiera się na implementacji klasy dla automatu wydającego przekąski i napoje, a także na wdrożeniu wszelakich jego funkcjonalności jak np. wybieranie pozycji, sprawdzanie/zmienianie ceny produktu, wydawanie artykułu po uzyskaniu wpłaty, zwrot pieniędzy podczas anulowania żądania, wprowadzanie nowych produktów na poszczególne pozycje oraz specyfikacja ich parametrów.

# Project description

The project is based on design and implement class for a vending machine (dispensing snacks and drinks). Moreover project includes implementation of all its functionalities such as selecting items, checking/changing the price of product, dispensing product after getting a certain amount of coins, refunding when request is cancelled, adding new products and specification of their parameters.

# Instrukcja użytkownika

Program został wykonany jako pseudo-graficzny symulator użytkowania automatu wydającego przekąski oraz napoje. Oznacza to, iż za pomocą znaków graficznych został odwzorowany ogólny wygląd maszyny, który zmienia się dynamiczne wraz z przeprowadzanymi przez klienta/konserwatora interakcjami.

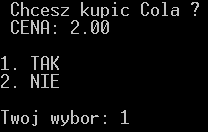


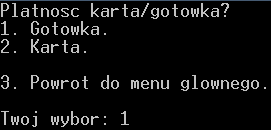
***Główne menu*** programu opiera się na 3 opcjach:

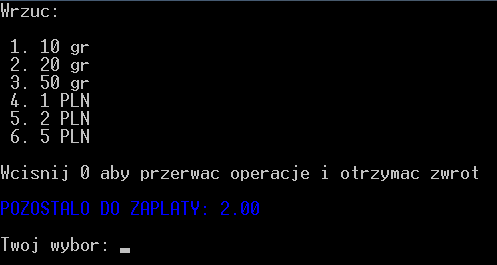
* ***Wyświetlanie automatu*** (w każdej chwili użytkownik może sprawdzić stan, cenę oraz liczbę wszystkich dostępnych w automacie produktów jednocześnie dzięki pseudo-graficznemu przedstawieniu maszyny.)
* ***Zakup produktu*** (użytkownik ma możliwość wyboru produktu, a następnie przeprowadzany jest przez kolejne fazy transakcji.)
* ***Konfiguracja pól maszyny.*** (Opcja symulująca dla konserwatora urządzenia, który otrzymuje w tym menu dodatkowe przywileje pozwalające na ingerencje w parametry każdego produktu tj. nazwa / cena / liczba dostępnych sztuk.

Dalsza zasada działania symulatora opiera się na klarownym przeprowadzeniu użytkownika przez kolejne fazy transakcji. Dzięki zastosowaniu klasy automat oraz jej obiektów w łatwy sposób zostały powiązane pola automatu wraz z ich zawartością.

Przykład przeprowadzania transakcji:







W tej części działania programu została zrealizowana symulacja wrzucania monet przez klienta oraz reakcji automatu na dany typ monety. Dla przykładu, jeżeli dany produkt kosztuje 2 zł, a użytkownik wrzuci do automatu monety: 1zł, 50gr, a następnie 5zł, to zostanie wywołana procedura zwrotu nadwyżki, a klient otrzyma zwrot wysokości 4.50zł.

C:\Users\Komputer\Desktop\Screenshot_10.png

Po wywołaniu metody odpowiedzialnej za wydawanie reszty, wywoływana jest metoda tej samej klasy odpowiedzialna za wydanie produktu.

Na koniec następuje sekwencja pożegnalna, a automat wraca do stanu początkowego jednocześnie odpowiednio zmniejszając liczbę dostępnych produktów.

# Kompilacja

Do uruchomienia i w pełni funkcjonalnego działania programu wystarczająca jest standardowa kompilacja.

Projekt został stworzony oraz testowany na systemie operacyjnym Windows 7.

# Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

Automat.h – deklaracja oraz implementacja klasy Automat

Modyfikacja.h – menu techniczne klasy Automat

Wyswietlanie.h – pseudo-graficzna prezentacja obiektów klasy Automat

Źródło.cpp - menu główne

# Zależności

Brak

# Opis klas

W projekcie utworzono następującą klasę:

* Automat – reprezentuje maszynę wydającą przekąski i napoje.

Dla klasy Automat utworzono następujące metody:

* void Wybor\_produktu(void) - wybór produktu i formy płatności.
* void Wydaj\_produkt(void) – wydawanie wybranego produktu.
* void Wydaj\_reszte(double reszta) - wydawanie reszty klientowi.
* void Wplata\_gotowki(void)- symulacja wpłaty, obliczanie należnej reszty.
* void Platnosc\_karta(void) - symulacja wpłaty za pomocą karty płatniczej.
* void Modyfikuj(void) - metoda dostępna dla konserwatora pozwalająca ingerować bezpośrednio w stan oraz parametry produktów.

Metody oraz settery pozwalające na modyfikacje obiektów:

* void Dodaj\_Usun\_Produkt(void)- dodawanie/usuwanie produktów
* void set\_nazwa(const string& nazwa)- ustawianie nazwy pod konkretnym polem
* void set\_cena(const double& cena) ustawianie ceny pod konkretnym polem
* void set\_liczba\_dostepnych(const int& liczba\_dostepnych)- jw.
* void set\_numerek(const int& numerek) - ustawienie numerów pól w automacie

# Zasoby

Brak

# Dalszy rozwój i ulepszenia

Dalszy rozwój projektu postrzegany jest przeze mnie w dwóch kategoriach:

* ***Dalszy rozwój symulatora****.*

Kontynuacja pracy nad symulacją pracy automatu oraz ewentualne powiązanie jego funkcjonalności z np. terminalem, który w następnej fazie łączy się z bankiem i odprowadza określoną kwotę z konta klienta.

Możliwe byłoby także rozszerzenie zadań maszyny na np. wydawanie gorących napoi wraz z dodatkami określanymi przez użytkownika (kawa z mlekiem/bez mleka, mało/dużo cukru itp.)

* ***Przygotowanie projektu pod implementacje jako oprogramowanie dla prawdziwego automatu.***

Koncepcja polegająca na wyeliminowaniu funkcji symulacyjnych, a następnie przygotowanie kodu pod zaprogramowanie mikrokontrolera obsługującego mechaniczne elementy automatu potrzebne do wydawania produktów/reszty, przyjmowanie monet, wyświetlania komunikatów na wyświetlaczu LCD.

# Inne

Brak