



POLITECHNIKA KOSZALIŃSKA
WYDZIAŁ ELEKTRONIKI I INFORMATYKI



Zastosowania programowania obiektowego

Projekt

Temat: Automat z przekąskami

Wykonujący projekt:
Marcel Pietrzak U-18258

Koszalin, Styczeń 2022

Oprogramowanie używane przy tworzeniu programu:

System operacyjny Windows 10 Home 21H1 ver.19043.1466

IDE używany do napisania programu : Visual Studio Community 2022 ver.17.0.5

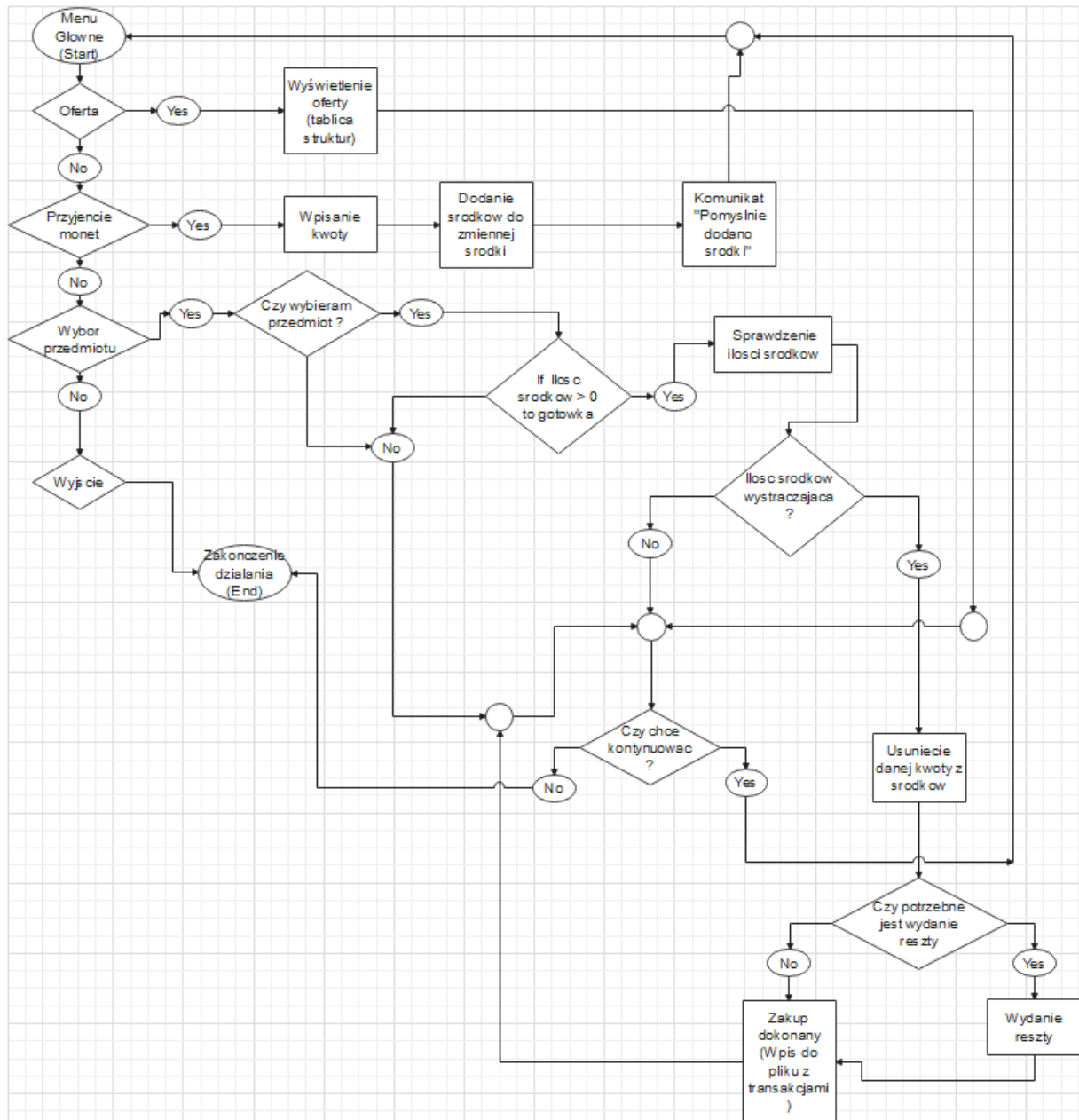
Narzędzia kompilacji : MSVC ver.143 – VS2022 dla architektury x64/x86 z pakietem C++/CLI w wersji 143

.NET w wersji 6.0.0 oraz 4.7.2

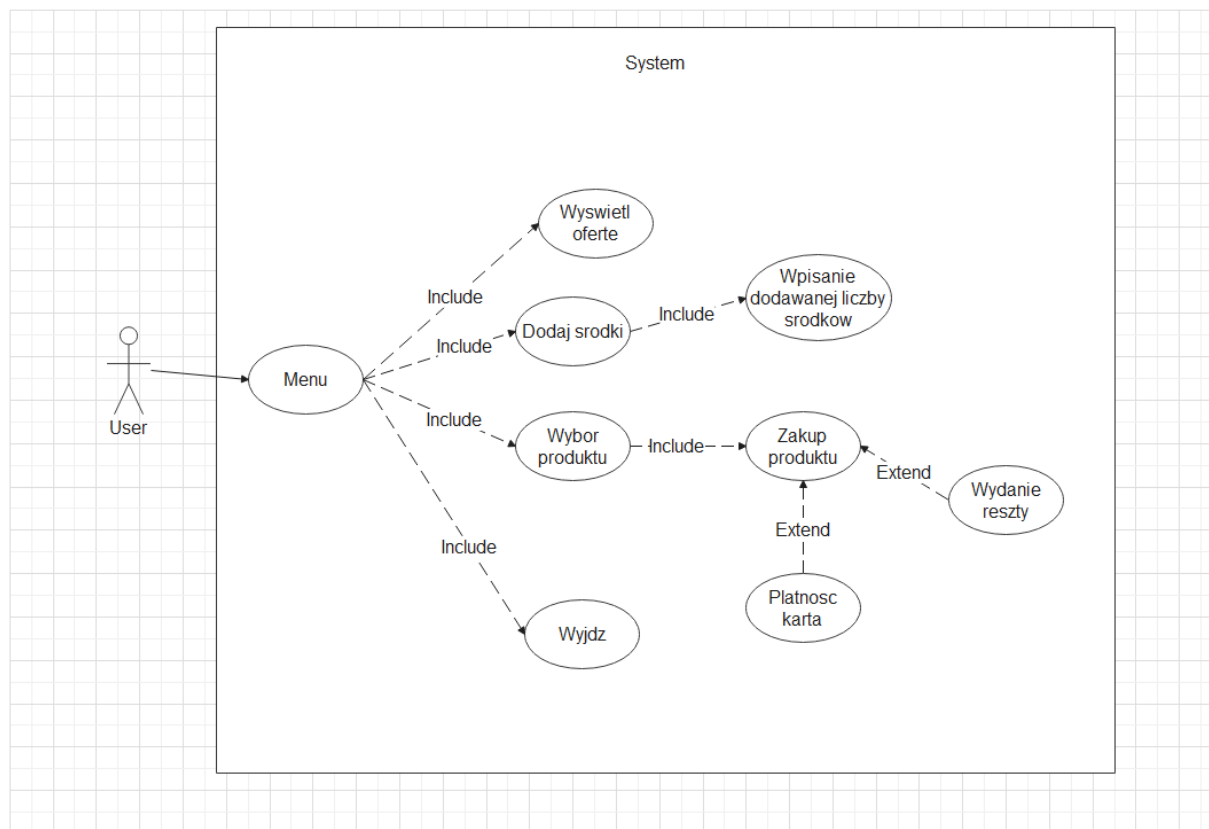
1.Opis programu

Program to symulator „Automatu z przekąskami” . Jak w prawdziwym automacie sprzedającym przekąski można do niego wprowadzać środki o nominałach polskich monet 5zł,2zł,1zł,0.5zł,0.2zł,0.1zł automat wybraniu przedmiotu za mniejszą ilość niż wprowadzone środki potrafi wydać resztę również w tych samych nominałach co wprowadzane środki. Automat może wyświetlać liste produktów oraz ich cen, jest to symulacja widoku produktów za szybą w prawdziwym automacie. Użytkownik posiada możliwość płatności kartą za produkt. Wyświetla on również komunikat o zakupie produktu. Program w wersji konsolowej posiada również możliwość zapisu transakcji z rozróżnieniem rodzaju wybranej metody płatności.

2. Blokowy schemat logiki programu

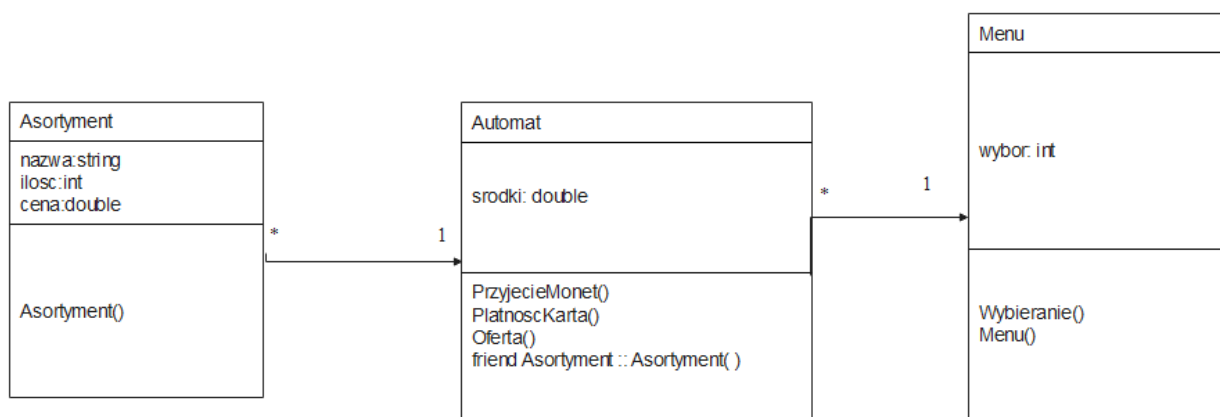


3.Use Case Diagram

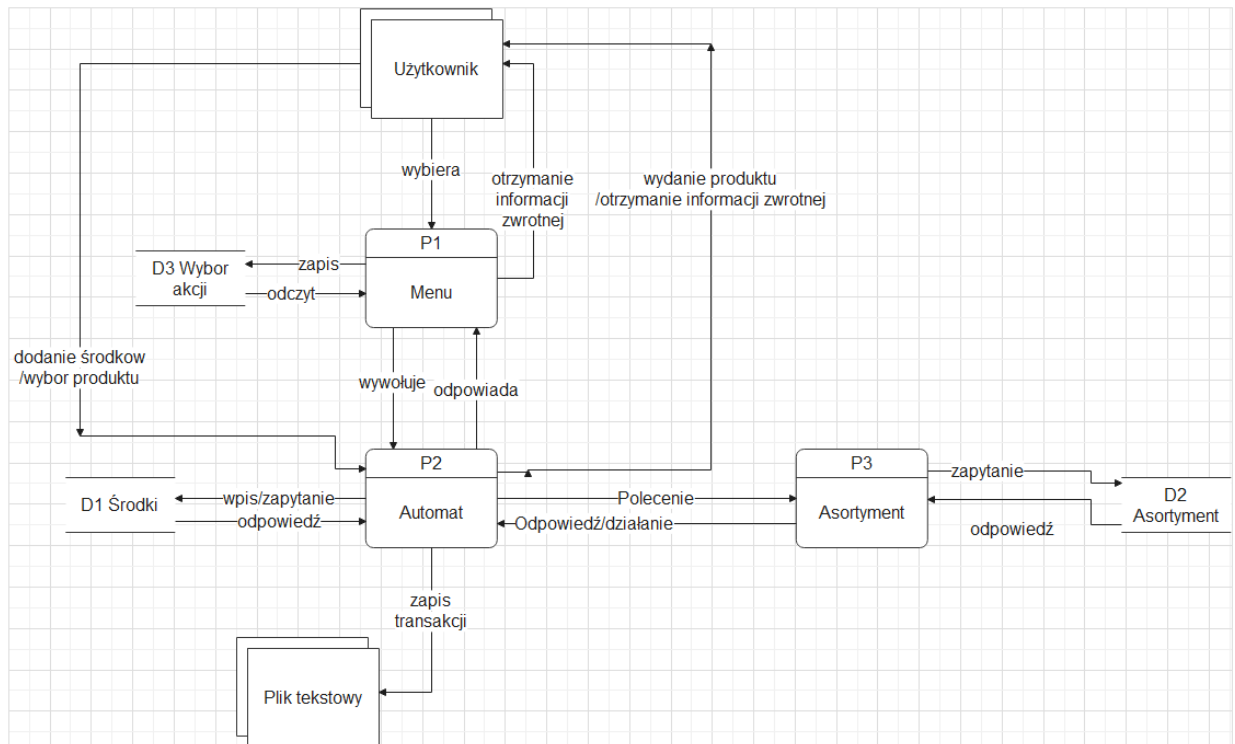


4.Diagram Klas

Class Diagram



5. Diagram przepływu danych



6.Kod programu w wersji konsolowej

```
#include <iostream>
#include "funkcje.h"
#include "menu.cpp"
using namespace std;

int main()
{
    menu x; // utworzenie obiektu x
    x.odpal(); // wywołanie funkcji odpal na obiekt x
    return 0;
}
```

6.1 Zawartość pliku main.cpp

```
#include "funkcje.h"
#include <iostream>
#include "automat.cpp"
using namespace std;

void menu::odpal() {
    automat c;
    int opcje = 0;
    cout << "=====Witaj===== " << endl;
    cout << "Co chcesz zrobić ? " << endl;
    cout << "1 = oferta , 2 = przyjęcie monet, 3 = wybór" << endl;
    cin >> opcje; // po wpisaniu zmiennej opcje odpalana jest funkcja która obsługuje daną funkcjonalność automatu
    if (opcje <= 3) {
        switch (opcje) {
            case 1:
                c.oferta(); // funkcja odpowiedzialna za wyświetlenie oferty
                break;
            case 2:
                c.przyjecie_monet(); // funkcja odpowiedzialna za przyjęcie monet po zakończeniu dodawania
                break; // środków następuje zasycenie o kontynuowanie transakcji (1 funkcjonalność wyboru, inne koniec i zwrócenie środków)
            case 3:
                c.wybieranie(); // funkcja odpowiedzialna za wybieranie produktu do kupna
                break; // wraz z tym dodanie środków wcześniej automatycznie wybierana
                // jest funkcjonalność płatności kartą
            }
        }
        cout << "Error wybór poza zakres" << endl;
    }
};
```

6.2 Zawartość pliku menu.cpp

```
#ifndef FUNKCJE_H_INCLUDED
#define FUNKCJE_H_INCLUDED
#pragma once
#include <iostream>
#include <string>

using namespace std;

class menu {
public:
    void odpal(); // funkcja obsługująca działanie menu
};

struct aso { // deklaracja struktury
    string nazwa;
    int ilosc;
    double cena;
};
```

6.3 Zawartość pliku funkcje.h

```

class asort{
public:
    aso arr[30] = { // deklaracja tablicy (z uzyciem struktury) ktora przechowuje
        {"chipsy_paprykowe",10,2.5}, {"chipsy_cebulowe",10,2.5},
        {"chipsy_solone",10,2.5}, {"chipsy_BBQ",10,2.5},
        {"Woda_niegazowana",15,2.00}, {"woda_gazowana",10,2.00},
        {"orzeczki_solone",10,2.5}, {"orzeczki_karmelowe",10,2.5},
        {"batonik_czekoladowy",10,2.5}, {"batonik_toffie",10,2.5},
        {"batonik_energetyczny",10,2.5}, {"napój_energetyczny",10,2.5},
        {"herbata_mrozona",10,2.5}, {"Oranżada_czerwona",10,2.5},
        {"Oranżada_niebieska",10,2.5}, {"napój_izotoniczny",10,2.5},
        {"orzeczki_w_czekoladzie",10,2.5}, {"suszone_jablka",10,2.5},
        {"beef_jerky",10,2.5}, {"zelki",10,2.5},
        {"prazona_kukurydza",10,2.5}, {"slonecznik_prazony",10,2.5},
        {"sok_pomarancowy",10,2.5}, {"sok_jablkowy",10,2.5},
        {"sok_winogronowy",10,2.5}, {"sok_multiwitamina",10,2.5},
        {"drazetki_kokosowe",10,2.5}, {"drazetki_czekoladowe",10,2.5},
        {"batonik_mleczny",10,2.5}, {"batonik_advocat",10,2.5}
    };
};

```

6.4 Zawartość pliku funkcje.h

```

class automat{
private:
    double srodki = 0.0; // zmienna przechowujaca srodki
    string koniec; // zmienna obslugujaca wyjscie z petli while
    int moneta; // zmienna obslugujaca wybor monet jakie zostana dodane do srodkow (switch case)
    int wybor; // zmienna obslugujaca wybor numeru produktu
    int i; // iterator po ktorym chodza pentle
public:
    void karta(); // funkcja obslugujaca platnosc karta
    void oferta(); // funkcja odpowiedzialna za wywiztlanie oferty
    void przyjecie_monet(); // funkcja odpowiedzialna za przyjecie monet
    void wybieranie(); // funkcja odpowiedzialna za wybieranie produktu do kurna
};

```

```
#endif // FUNKCJE_H_INCLUDED
```

6.5 Zawartość pliku funkcje.h

```

#include "funkcje.h"
#include <iostream>
#include <string>
#include <stdlib.h>
#include <fstream>
using namespace std;

asort b; // tworzenie obiektu b klasy asort
menu c; // tworzenie obiektu c klasy menu

void automat::oferta(){
    string powrot;
    cout << "=====PRZEKASKI===== " << endl;
    for(int i = 0; i < 30; i++){ // petla przechodzaca po asortymencie
        cout << "Produkt" << "[" << i << "]" = " << b.arr[i].nazwa << "Ilosc = " << b.arr[i].ilosc << "Cena = " << b.arr[i].cena << endl;
    };
    cout << "===== " << endl;
    cout << "Powrot ? y = tak / inna akcja = nie " << endl;
    cin >> powrot;
    if(powrot == "y"){
        system("CLS");
        c.odpal();
    };
};

```

6.6 Zawartość pliku automat.cpp


```

void automat::przyjecie_monet(){ //Funkcja przyjmująca monety do automatu
    int kon;
    srodki = 0.0;
    koniec = "Nie";
    while(koniec == "Nie" || koniec == "nie" || koniec == "Nię"){
        cout << "Jakie monety włożysz ?" << endl;
        cout << " 1 = 1zł , 2 = 2zł , 5 = 5zł , 20 = 0.2zł , 10 = 0.1zł , 50 = 0.5zł , 99 - stan środków" << endl;
        cin >> moneta; //prziancie wartosci do dodania do srodkow
        switch(moneta){
            case 1 : // case do wybrania po wybraniu poprzez wprowadzenie zmiennej wybor dodaje odpowiednia kwote do zmiennej srodki
                srodki = srodki + 1;
                cout << "Posiadasz : " << srodki << " zł" << endl;
                break;
            case 2 :
                srodki = srodki + 2;
                cout << "Posiadasz : " << srodki << " zł" << endl;
                break;
            case 5 :
                srodki = srodki + 5;
                cout << "Posiadasz : " << srodki << " zł" << endl;
                break;
            case 20 :
                srodki = srodki + 0.2;
                cout << "Posiadasz : " << srodki << " zł" << endl;
                break;
            case 10 :
                srodki = srodki + 0.1;
                cout << "Posiadasz : " << srodki << " zł" << endl;
                break;
            case 50 :
                srodki = srodki + 0.5;
                cout << "Posiadasz : " << srodki << " zł" << endl;
                break;
            case 99 :
                cout << srodki << endl;
                break;
        }
    }
}

```

6.7 Zawartość pliku automat.cpp

```

        default:{cout << "Nieprawidłowa wartość" << endl;}
    }
    cout << "Koniec ?" << endl;
    cin >> koniec;
}
cout << "Twoje środki to : " << srodki << endl;
cout << "Czy chcesz kontynuować transakcję ? 1-tak " << endl;
cin >> kon;
if(kon == 1){// if odpowiadający zak kontynuowania transakcji lub zwrocenia srodkow
    wybieranie();
}else{
    cout << "Automat wydał twoje środki : "<< srodki << "zł" << endl; // Komunikat symulujący wyrzut monet w przypadku braku zakupu
}
};

```

6.8 Zawartość pliku automat.cpp

```

void automat::wybieranie(){
    cout << " Ilosc srodkow " << srodki << endl;
    cout << "Wybierz : " << endl;
    cin >> wybor; // zmienna wybor z automatu
    if( wybor<30){
        if(srodki > 0){
            int R,P; //zmienna R i P obslugujace algorytm wyliczalacy kasze
            R = (srodki*100) - (b.arr[wybor-1].cena*100); //wyzalczanie wartosci ktora nalezy zwrocic
            int N[6]={500, 200, 100, 50, 20, 10}; // tablica przechowujaca nominaly (rozniecia na kwotach)
            i=0;
            while(R>0){
                if(R >= N[i]){ // sprawdzanie czy R ktora ma zostac zwroczona jest wieksza lub rowna nominalowi ktory mozna wykras
                    P=R / N[i]; // p ma wyznaczone wartosc R podzielona przez wartosc nominalu
                    R=R-(N[i]*P); // R wartosc zwroczona to wartosc zwroczonego o dany nominali widujemy dane ilosc razy
                    cout << "Bazeta : " << endl;
                    cout << N[i]/100.0 << "x" << P << " zł" << endl; // wyliczenie kaszy
                }
                i++;
            }
            ofstream Plik("zapis.txt",ios::app); // otwarcie strumienia zapisu do pliku
            Plik << "Kupione " << b.arr[wybor].nazwa << " za " << b.arr[wybor].cena << " platnosc gotowka " << endl; // wpisanie informacji o transakcji do pliku z uwzględnieniem rodzaju
            Plik.close(); // zamknięcie strumienia zapisu do pliku
            cout << "Wybrano : " << b.arr[wybor].nazwa << endl; // wyliczenie komunikatu o nazwie zakupionego towaru (symulacja wybrania towaru z automatu)
            b.arr[wybor].ilosc = b.arr[wybor].ilosc - 1; // odejście 1 sztuki zakupionego towaru z automatu
        }else{
            cout << "Platnosc karta" << endl;
            karta(); // wywołanie funkcji platnosc karta
        }
    }else{
        cout << "Nieprawidlowa pozycja" << endl; // komunikat w przypadku wpisania nieprawidlowej pozycji z automatu
    }
}
};

```

6.9 Zawartość pliku automat.cpp

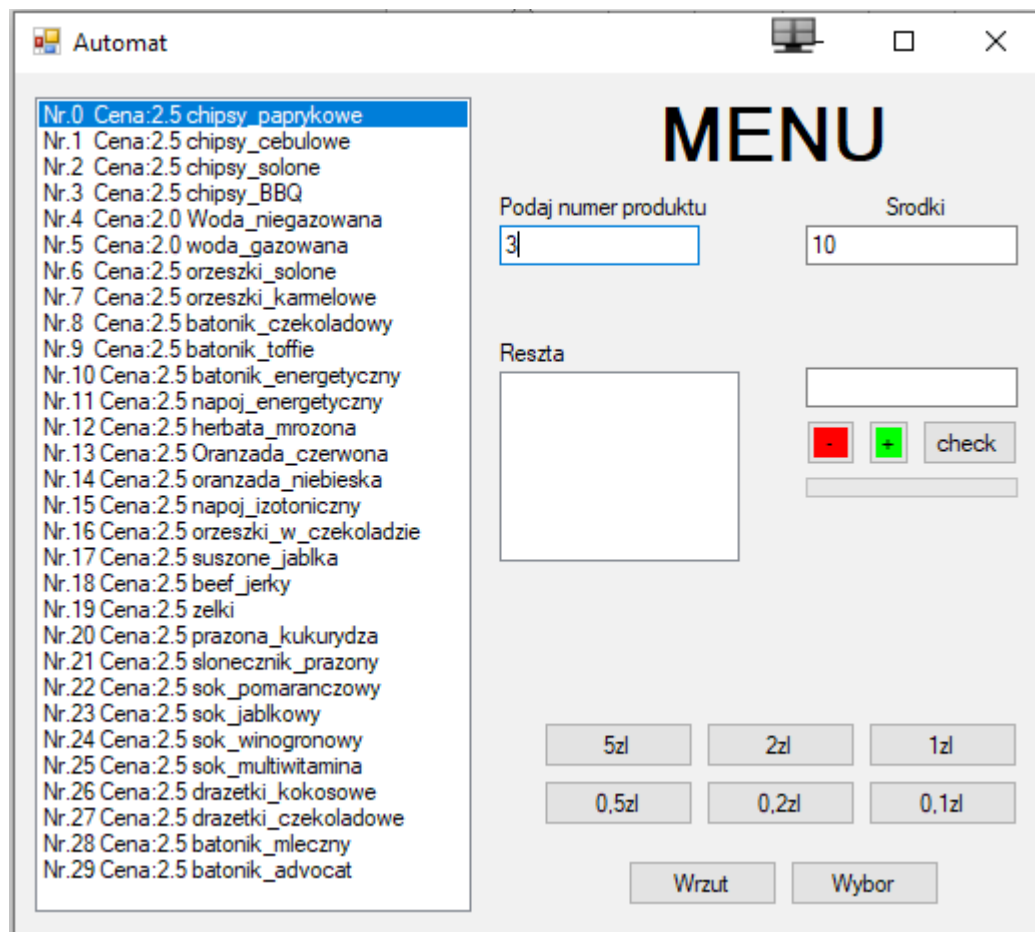
```

void automat::karta(){ //funkcja obsluguje uproszczony mechanizm platnosci karta
int przyl; // zmienna przechowujaca czy karta zostala przylozona
cout << "Przyloz karte 1 - tak inna liczba nie" << endl;
cin >> przyl; // wnie o tym czy karta zostala przylozona
if(przyl == 1 ){ // funkcjonalnosc odpalana w przypadku przylozenia karty (zmienna przyl = 1)
    cout <<"Platnosc karta" << endl;
    cout <<"Ladowanie ..." << endl;
    cout << "." << endl;
    cout << "." << endl;
    cout << "." << endl;
    cout << "Platnosc przyjeta"<< endl;
    cout << "wybrano " << b.arr[wybor].nazwa << endl; // Wypisanie produktu po platnosci
    b.arr[wybor].ilosc = b.arr[wybor].ilosc - 1; // usuniecie 1 sztuki z asortymentu
    ofstream Plik("zapis.txt", ios::app); //otwarcie strumienia zapisu transakcji do pliku
    Plik << "Kupiono " << b.arr[wybor].nazwa << " za " << b.arr[wybor].cena << " platnosc karta " << endl;
    Plik.close(); // zamknienie strumienia do zapisu pliku
}else{
    cout <<"Koniec plantosci"<<endl;
    c.odpal(); // ponowne wywołanie menu
};
}
}

```

6.10 Zawartość pliku automat.cpp

7. Graficzny interfejs



Automat

MENU

Podaj numer produktu:

Srodki:

Reszta:

Nr.0 Cena:2.5 chipsy_paprykowe
Nr.1 Cena:2.5 chipsy_cebulowe
Nr.2 Cena:2.5 chipsy_solone
Nr.3 Cena:2.5 chipsy_BBQ
Nr.4 Cena:2.0 Woda_niegazowana
Nr.5 Cena:2.0 woda_gazowana
Nr.6 Cena:2.5 orzeszki_solone
Nr.7 Cena:2.5 orzeszki_kamelowe
Nr.8 Cena:2.5 batonik_czekoladowy
Nr.9 Cena:2.5 batonik_toffie
Nr.10 Cena:2.5 batonik_energetyczny
Nr.11 Cena:2.5 napoj_energetyczny
Nr.12 Cena:2.5 herbata_mrozona
Nr.13 Cena:2.5 Oranzada_czerwona
Nr.14 Cena:2.5 oranzada_niebieska
Nr.15 Cena:2.5 napoj_izotoniczny
Nr.16 Cena:2.5 orzeszki_w_czekoladzie
Nr.17 Cena:2.5 suszone_jablka
Nr.18 Cena:2.5 beef_jerky
Nr.19 Cena:2.5 zelki
Nr.20 Cena:2.5 prazona_kukurydza
Nr.21 Cena:2.5 slonecznik_prazony
Nr.22 Cena:2.5 sok_pomarancowy
Nr.23 Cena:2.5 sok_jablkowy
Nr.24 Cena:2.5 sok_winogronowy
Nr.25 Cena:2.5 sok_multiwitamina
Nr.26 Cena:2.5 drazetki_kokosowe
Nr.27 Cena:2.5 drazetki_czekoladowe
Nr.28 Cena:2.5 batonik_mleczny
Nr.29 Cena:2.5 batonik_advocat

7.1 Okno programu w wersji z interfejsem graficznym.