

Podstawowe kontrolki danych ASP.NET

Cele ćwiczenia

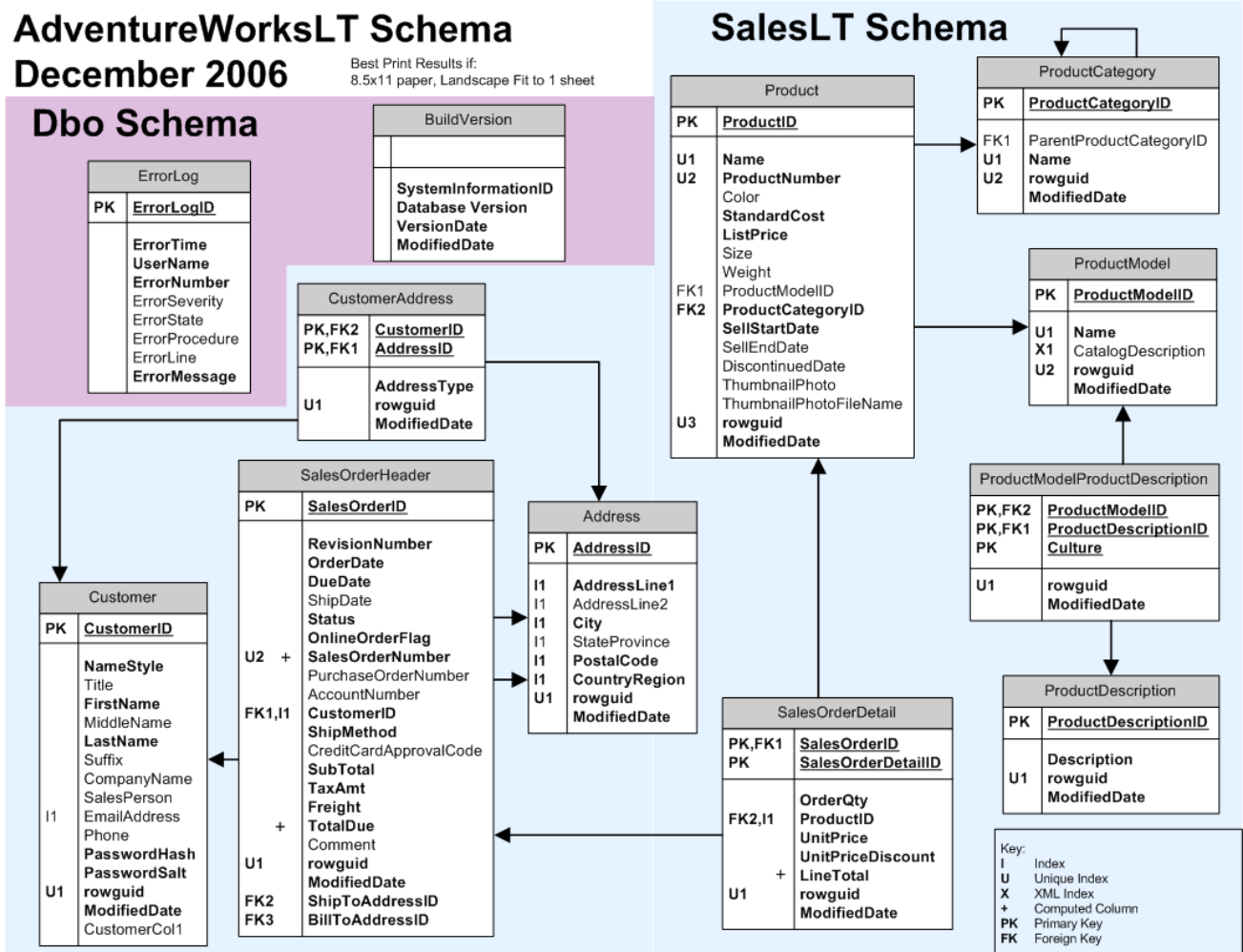
- Zapoznanie z podstawowymi kontrolkami danych ASP.NET i sposobami ich konfiguracji.
- Zapoznanie z metodami konfiguracji źródeł danych i tworzenia zapytań SQL przy pomocy narzędzi Visual Studio.
- Zapoznanie ze sposobami przekazywania parametrów do zapytań SQL.

Przykładowe aplikacje znajdują się pod adresami:

- <http://argo.umg.edu.pl/www/AdventureWorksLT/>
- <http://argo.umg.edu.pl/www/aplikacje/cw09/>

Baza danych AdventureWorksLT

Baza danych *AdventureWorksLT* jest przykładową bazą danych *OpenSource* przeznaczoną do nauki programowania, udostępnioną na stronie <https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks>. Diagram bazy danych *AdventureWorksLT* przedstawiony jest na rysunku 1.



Rys. 1. Diagram bazy danych *AdventureWorksLT*

Konfiguracja połączenia z bazą danych *AdventureWorksLT*

Wariant 1 – korzystanie z zewnętrznego serwera MS SQL Server.

W oknie *Server Explorer* dodaj nowe połączenie MS SQL Server. Konfiguracja połączenia jest przedstawiona na rysunku 2:

- Nazwa serwera: *argo.umg.edu.pl*,
- Uwierzytelnianie: *Use SQL Server Authentication*,
- User = *student*, Password = *student*,
- zaznacz: *Zapisz moje hasło* (*Save my password*),
- wybierz bazę danych: *AdventureWorksLT*.

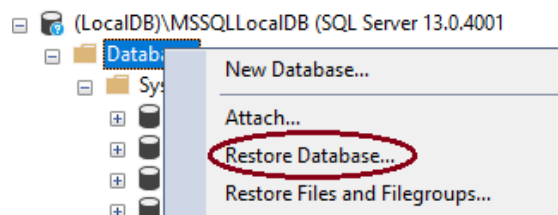
Rys. 2. Konfiguracja połączenia z serwerem MS SQL Server *argo.umg.edu.pl*

Wariant 2 – lokalna baza danych.

1. Pobierz ze strony <https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks> właściwy plik z archiwum bazy danych *AdventureWorksLT* i zainstaluj go na swoim serwerze lokalnym – (LocalDB)\MSSQLLocalDB. Do instalacji należy użyć programu *Microsoft SQL Server Management Studio*. Dane do logowania na rysunku 3.

Rys. 3. Dane logowania na lokalnym serwerze MS SQL Server

2. Odtwórz bazę danych *AdventureWorksLT* z pobranego pliku na lokalnym serwerze SQL, rysunek 4.



Rys.4. Opcja odtwarzania bazy danych

Lokalna kopia bazy danych pozwoli na tworzenie aplikacji na komputerze bez dostępu do serwera SQL *argo.umg.edu.pl*, np. w domu. Po przeniesieniu aplikacji na serwer zmiany źródła

danych można dokonać zmieniając łańcuch połączenia źródła danych w pliku *web.config*. Aby ułatwić zmianę źródła danych, należy korzystać z jednego łańcucha połączenia do bazy danych *AdventureWorksLT* w całej aplikacji.

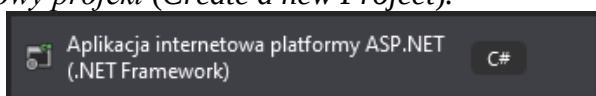
Utworzenie nowego projektu aplikacji internetowej w Visual Studio

Najlepiej korzystać z Visual Studio 2019, z zainstalowanymi komponentami



Korzystanie z Visual Studio 2022 wymaga dodatkowej instalacji komponentów i zmiany ustawień domyślnych.

1. Uruchom *Visual Studio*. Nazwy poleceń w instrukcji są napisane po polsku i po angielsku, aby ułatwić korzystanie z obu wersji językowych *Visual Studio*.
2. *Utwórz nowy projekt (Create a new Project)*.



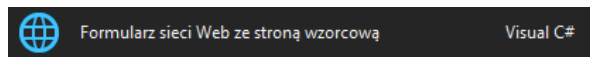
3. Wybierz: *Aplikacja internetowa platformy ASP.NET (.NET Framework) C#* → *Next* – *ASP.NET Web Application (.NET Framework) C#* → *Next*
4. W oknie *Konfiguruj nowy projekt (Configure your new Project)* możesz wpisać nazwę aplikacji i wybrać wersję platformy .NET Framework (np. 4.7.2) → *Utwórz (Create)*.
5. Na stronie *Utwórz nową aplikację internetową platformy ASP.NET (Create a new ASP.NET Web Application)* wybierz *Web Forms* i odznacz *Konfiguruj dla protokołu HTTPS (Configure for HTTPS)* → *Utwórz*.

Pierwsza strona aplikacji – Lista klientów

Strona *Lista klientów* będzie prezentowała dane klientów sklepu. Dane osobowe klientów umieszczone są w tabeli *Customer* a ich adresy w tabeli *Address*. Tabela *CustomerAddress* realizuje związek wiele do wielu między tabelami *Customer* i *Address*. Jako pierwsza na stronie zostanie umieszczona kontrolka *DropDownList*, w której zostaną umieszczone skrócone dane (nazwisko + imiona) wszystkich klientów. Dane klientów powinny być uporządkowane alfabetycznie.

Dodaj do aplikacji nową stronę.

- W oknie *Eksplorator rozwiązań (Solution Explorer)* kliknij prawym klawiszem myszki na katalog główny lub dodany przez siebie.
- Z menu kontekstowego wybierz: *Dodaj* → *Nowy element...* → *Formularz sieci Web ze stroną wzorcową* → *Dodaj (Add)* → *New Item* → *Web Form with Master Page* → *Add*.

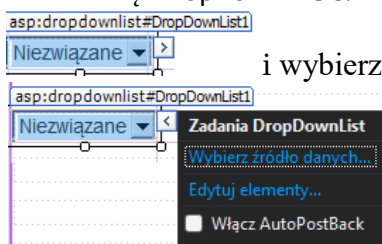


- W oknie *Wybierz stronę wzorcową (Select a Master Page)* kliknij *Side.Master* → *OK*.

Konfiguracja kontrolki DropDownList

1. Na stronie, w widoku *Projekt (Design)*, umieść kontrolkę *DropDownList*.

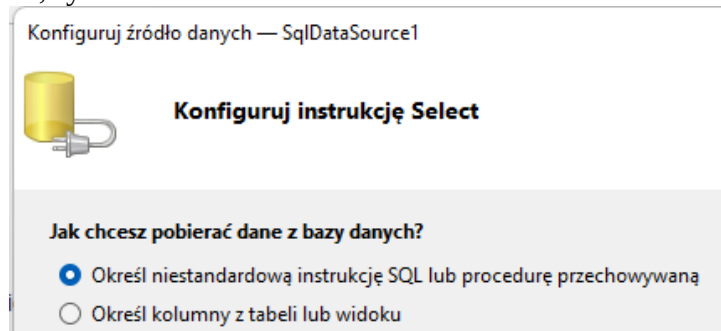
2. Kliknij „>” w prawym górnym rogu kontrolki i wybierz z Menu pozycję:



Wybierz źródło danych (Choose Data Source).

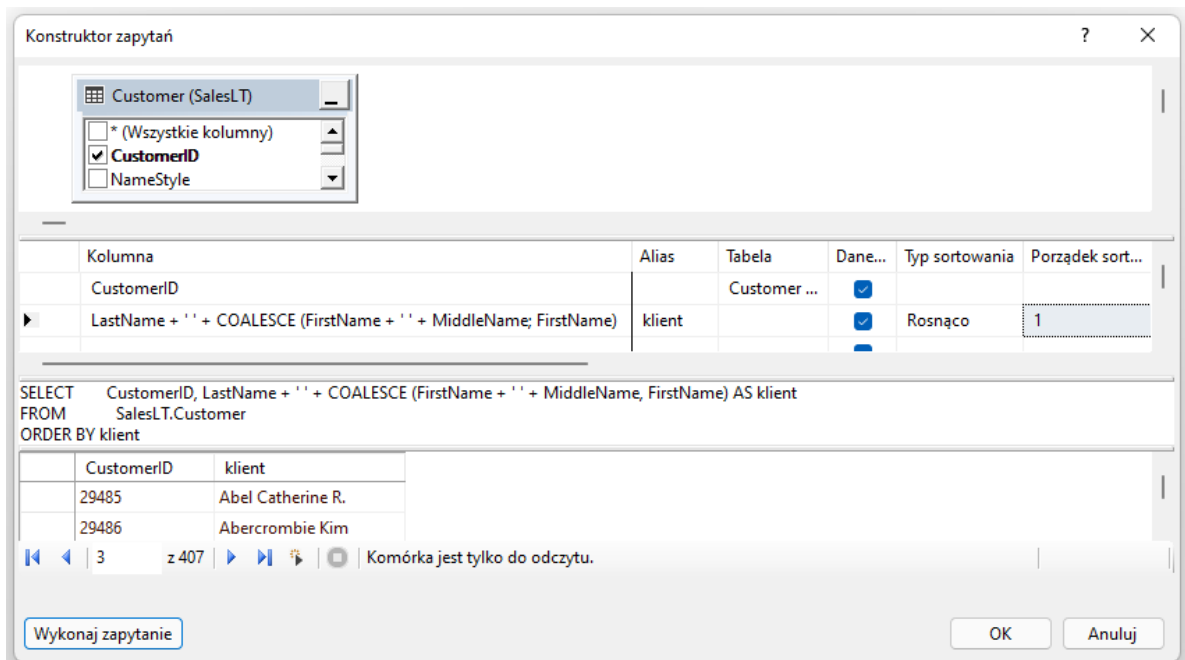
3. Skonfiguruj nowe źródło danych:

- Wybierz źródło danych = <Nowe źródło danych...> (Select a data source = <New data source...>),
- Skąd aplikacja będzie pobierać dane? = Baza danych (Where will the application get data from? = Database) → OK,
- Którego połączenia danych ma używać aplikacja do łączenia się z bazą danych? (Which data connection should your application use to connect to the database?) kliknij Nowe połączenie (New Connection).
- Skonfiguruj połączenie z serwerem albo lokalną bazą danych AdventureWorksLT.
- Next,
- W oknie Konfiguruj źródło danych wybierz: Specify a custom SQL statement or stored procedure → Next, rysunek 5.



Rys. 5. Wybór sposobu pobierania danych

- Kliknij przycisk *Konstruktor zapytań...* (Query Builder),
 - dodaj tabelę *Customer*,
 - utwórz dwukolumnową kwerendę:
 1. kolumna: *CustomerID*,
 2. kolumna: *LastName + ' ' + COALESCE (FirstName + ' ' + MiddleName, FirstName)*, rysunek 6.
 - zamknij *Konstruktor zapytań*.

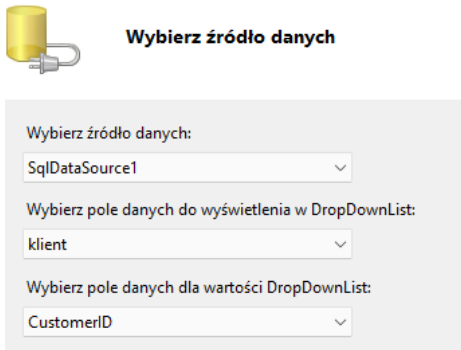


Rys.6. Zapytanie w konstruktorze zapytań

- Zakończ konfigurację źródła danych.
4. W oknie *Data Source Configuration Wizard* wybierz jak na rysunku 7:

- w polu *Select a data field to display in the DropDownList*: 2 kolumnę (jakby się nie nazywała),
- w polu *Select a data field for the value of the DropDownList*: 1 kolumnę *CustomerID*.

Kreator konfiguracji źródła danych



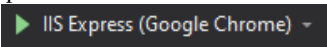
Wybierz źródło danych

Wybierz źródło danych:
SqlDataSource1

Wybierz pole danych do wyświetlenia w DropDownList:
klient

Wybierz pole danych dla wartości DropDownList:
CustomerID

Rys. 7. Konfiguracja sposobu wyświetlania danych w *DropDownList*

5. Zapoznaj się z efektem pracy uruchamiając Debugowanie strony .
6. Dopracuj konfigurację źródła danych kontrolki.
 - Dodaj spacje między nazwiskiem i imionami.
 - Usuń pozycje NULL z wyniku zapytania, które pojawiają się, gdy *MiddleName* jest NULL. W tym celu można wykorzystać funkcję *COALESCE(x1, x2, ...)* zwracającą pierwszy argument, który jest różny od NULL. Funkcja *COALESCE* jest używana przy łączeniu łańcuchów, z których jeden lub więcej może być NULL, np. *COALESCE (FirstName + MiddleName; FirstName)*; uwaga w *Konstruktorze zapytań* argumenty funkcji rozdziela się średnikami.

Prezentacja szczegółowych danych o kliencie

1. Na stronie umieść kontrolkę *DetailsView*.
2. Wybierz *Konfiguruj źródło danych* = *<Nowe źródło danych... >* (*Select a data source* = *<New data source...>*).
3. Skonfiguruj źródło danych tak, aby dane zawierały szczegółowe informacje o kliencie wybranym w kontrolce *DropDownList1*:
 - źródło danych = tabela *Customers*,
 - filtr: *CustomerID = @Param1*,
Parametr source = *Control*,
ControlID = *DropDownList1*.
4. Zapoznaj się z efektem swojej pracy,
 - aby umożliwić filtrowanie danych, dodaj do aplikacji przycisk lub w opcjach kontrolki *DropDownList1* zaznacz *Włącz AutoPostBack* (*Enable AutoPostBack*),
 - zmień *Auto Format* kontrolki *DetailsView1* i zobacz efekt zamiany.

Dodanie do strony informacji adresach klienta – kontrolka *GridView*

Do aplikacji dodaj kontrolkę *GridView* i skonfiguruj źródło danych tak, aby dane zawierały adresy klienta wybranego w kontrolce *DropDownList1*.

Dodanie do strony informacji zamówieniach klienta – kontrolka *GridView*

Do aplikacji dodaj kontrolkę *GridView* i skonfiguruj źródło danych tak, aby dane zawierały zamówienia klienta wybranego w kontrolce *DropDownList1*.

Druga strona aplikacji – Katalog produktów

Strona *Katalog produktów* powinna umożliwić użytkownikowi zapoznanie się z ofertą sklepu. Dane o produktach znajdują się w tabeli *Produkt*. Produkty podzielone są na kategorie, których nazwy znajdują się w tabeli *ProductCategory*. Kategorie podzielone są na podrzędne i nadrzędne, w kolumnie *ParentProductCategoryID* umieszczony jest identyfikator kategorii nadrzędnej. Kategorie, których *ParentProductCategoryID* jest NULL nie mają kategorii nadrzędnych (są korzeniami drzewa kategorii) i od nich rozpoczyna się poszukiwanie produktu.

Wybór kategorii nadrzędnej

1. Umieść na stronie kontrolkę `RadioButtonList1`.
2. Skonfiguruj źródło danych tak, aby wyświetlane dane zawierały listę kategorii nadrzędnych.

Wybór kategorii podrzędnej

1. Umieść na stronie kontrolkę `RadioButtonList2`.
 2. Skonfiguruj źródło danych tak, aby wyświetlane dane zawierały listę kategorii podrzędnych, filtr: `ParentProductCategoryID = RadioButtonList1.SelectedValue`.
- Hierarchia kategorii jest dwupoziomowa i nie trzeba dalej rozwijać drzewa kategorii.

Wybór produktu

1. Umieść na stronie kontrolkę `GridView1`.
2. Skonfiguruj źródło danych tak, aby wyświetlane dane zawierały nazwy produktów wybranej kategorii, filtr: `ProductCategoryID = RadioButtonList2.SelectedValue`.
3. W konfiguratorze kontrolki zaznacz opcję *Włącz wybieranie (Enabled Selection)* – zaznaczenie opcji umożliwi zastosowanie kontrolki do filtrowania danych.

Prezentacja szczegółowych danych produktu

1. Umieść na stronie kontrolkę `DetailsView1`.
2. Skonfiguruj źródło danych tak, aby wyświetlane dane zawierały szczegółowe dane o produkcie wybranym w kontrolce `GridView1`, filtr `ProductID = GridView1.SelectedValue`.
3. Do prezentacji produktu dodaj opis modelu, po angielsku lub w innym wybranym języku.

Synchronizacja kontrolek

Hierarchia filtrów danych kontrolek jest wielopoziomowa, co negatywnie wpływa na działanie strony, dlatego powinno się stronę dopracować tak, aby:

- po zmianie kategorii nadrzędnej widoczna była tylko ta kontrolka i kontrolka kategorii podrzędnych, bez wybranej kategorii,
- po zmianie kategorii podrzędnej, kontrolka wyboru produktu nie powinna mieć wybranej żadnej pozycji i żaden szczegółowy opis produktu nie powinien być widoczny.

W celu usunięcia wyboru pozycji w kontrolkach można wykorzystać instrukcje:

- `GridView1.SelectedIndex = -1;` – usuwa wybór w kontrolce `GridView1`,
- `RadioButtonList2.ClearSelection();` – usuwa wybór w kontrolce `RadioButtonList2`.

Instrukcje należy umieścić w funkcjach obsługi zdarzeń zmiany wyboru (indeksu) w kontrolkach radiowych.

Trzecia strona aplikacji – zakupy

Stronę *Katalog produktów* należy tak rozbudować, aby umożliwiała zbieranie zakupów do koszyka. Zawartość koszyka powinna być przechowywana jako parametr sesji w obiekcie klasy

System.Data.DataTable. Na listingu 1. przedstawiono kod tworzący obiekt klasy DataTable następnie wypełniający go kolumnami i wierszem danych.

Listing 1.

Kod tworzący obiekt klasy System.Data.DataTable , następnie wypełniający go kolumnami i wierszem danych

```
if (Session["kosz"] == null)
{
    DataTable kosz = new DataTable("kosz");
    // Dodanie do tabeli kolumny z autonumeracją
    DataColumn kolumnaTab;
    kolumnaTab = new DataColumn("Nr", Type.GetType("System.Int32"));
    kolumnaTab.AutoIncrement = true;
    kolumnaTab.AutoIncrementSeed = 1;
    kolumnaTab.AutoIncrementStep = 1;
    kolumnaTab.AllowDBNull = false;
    kolumnaTab.Unique = true;
    kosz.Columns.Add(kolumnaTab);
    // Dodanie do tabeli kolumn z danymi zamówienia
    kosz.Columns.Add("Nr towaru");
    kosz.Columns.Add("Nazwa towaru");
    kosz.Columns.Add("Ilość", Type.GetType("System.Int32"));
    // Zapisanie tabeli jako parametru sesji
    Session.Add("kosz", kosz);
}

// Dodanie zamówienia
DataTable zamowienia = (DataTable)Session["kosz"];
DataRow zamowienie;
zamowienie = zamowienia.NewRow();
zamowienie["Nr towaru"] = GridView1.SelectedDataKey.Value;
zamowienie["Nazwa towaru"] = GridView1.SelectedRow.Cells[1].Text;
zamowienie["Ilość"] = Convert.ToInt32(TextBox1.Text);
zamowienia.Rows.Add(zamowienie);
// Umieszczenie zawartości tabeli w kontrolce GridView2
GridView2.DataSource = zamowienia;
GridView2.DataBind();
```

Dane zamówienia *Nr towaru (ProduktID)* i *Nazwa towaru (Name)* są pobierane z kontrolki GridView1, a ilość sztuk z kontrolki TextBox1. Zebrane zamówienia są prezentowane w kontrolce GridView2.

Przebieg czynności

1. Dodaj do strony panel i umieść na nim przycisk i kontrolkę TextBox1.
2. Dodaj do strony kontrolkę GridView2 , będzie prezentować zawartość koszyka.
3. Dodaj obsługę kliknięcia przycisku dodającą do koszyka nową pozycję – kod wzorowany na listingu 2.
4. Tak skonfiguruj stronę, aby panel z kontrolkami dodania pozycji do zamówienia był widoczny tylko wtedy, gdy wybrany jest produkt w kontrolce GridView1. Można w tym celu wykorzystać kod z listingu 2.

Listing 2.

Kod ukrywający Panel1, gdy nie jest wybrana pozycja w kontrolce GridView1.

```
if (GridView1.SelectedIndex == -1)
    Panel1.Visible = false;
else Panel1.Visible = true;
```

Projekt

1. Dodać do aplikacji wybór klienta i zapisać wybranego klienta, jako parametr sesji.
2. Dodać do aplikacji stronę zawierającą dane zamówienia, czyli dane klienta i listę zakupionych towarów.