Autentificare. Obiecte principiale

Obiective

După completarea acestui laborator veți dobândi următoarele cunoștințe:

- Implementarea unui obiect principial personalizat.
- Setarea principiului pentru thread-ul curent.

Scenariu de laborator

Sunteți dezvoltator pentru o companie producătoare de biciclete Adventure Works. Ați fost însărcinați să realizați o aplicație pentru punctele de vânzare, astfel încât utilizatorii autorizați pot accesa produsele și pot returna banii clienților, iar utilizatorii neautorizați pot doar să interogheze stocul.

Ați identificat două roluri pentru utilizatorii autorizați: **Clerk** și **Manager**. Utilizatorii cu aceste roluri pot vinde obiecte, dar doar cei cu rolul **Manager** pot returna plăți clienților. Toți utilizatorii pot interoga stocul.

➢ Exercițiul 1- Crearea proiectului de instalare

În acest exercițiu veți observa codul oferit în suportul de laborator. Soluția din suportul de laborator conține o unitate de asamblare care simulează sistemul de inventar și permite utilizatorilor să achiziționeze și să returneze obiecte din inventar. Adițional, codul conține o infrastructură multidocument (**MDI**), care permite unui utilizator să se logheze ca un utilizator autentificat. Este furnizat codul pentru trei formulare: **Logon**, **Query** și **OrderForm**.

Taskul 1:

- Deschideți soluția oferită ca startup pentru laborator
- Observați cele două proiecte existente
  - **PointOfSale**
  - **Inventory**
- Expandați proiectul **PointOfSale** și faceți dublu clic pe **MainForm.cs**. Formularul se deschide în Forms Designer și este un formular **MDI**, care conține două obiecte **Authenticate** și **Action**.
- Realizați următoarele etape:
  - Faceți click pe **Authenticate**.

- Faceți dublu click pe **Login** – veți vizualiza codul metodei **loginToolStripMenuItem**, aceasta este metoda handler pentru evenimentul click al obiectului Login.

- Vizualizați codul din metodele **logoutToolStripMenuItem** și **queryToolStripMenuItem**.

Taskul 2: Clasa **Inventory**, **OrderForm** și **Query**

- În clasa MainForm.cs este un câmp numit inventory. Faceți click dreapta pe numele clasei **Inventory** și apoi pe **Go To Definition**.
  - Vizualizați codul clasei **Inventory**.

  Atenție! Această clasă este un obiect de acces la date simulate. Într-o aplicație reală datele sunt stocate într-o bază de date.

  - Vizualizați clasa **Product**.
    - Metodele **Increment** și **Decrement** afectează câmpul **StockLevel**.
    - Metoda **ToString** returnează numele produsului.

- Vizualizați controalele și codul ferestrei **Order Form**. Vizualizați următorul cod:
  - Bucla de inițializare a combo box-ului
  - Metoda **btnAdd_Click** – metoda handler pentru evenimentul Click a butonului >>>.
  - Metoda **btnOrder_Click** – metoda handler pentru evenimentul Click a butonului Order.
  - Metoda **btnRefund_Click** – metoda handler pentru evenimentul Click a butonului Refund.

- Vizualizați controalele și codul clasei **Query.cs**.

Taskul 3: Informații principiale

- Modificați metoda **UpdateUser** astfel:
  - Obțineți principiul de la thread-ul curent.
  - Adăugați cod pentru a formata un string cu informațiile de identitate și folosiți string-ul pentru a inițializa eticheta din josul form-ului.
  - Stringul ar trebui sa aiba formatul: „Name: <nume_identitate_aici>/ Authn: <daca_este_autentificat_sau_nu_aici>/ Roles: <rolurile_pe_care_le are_aici>”
  - **Reminder: Cele 3 roluri pe care vrem sa le verificam sunt „Clerk”, „Manager”. Un al treilea rol (pentru utilizatorii normali) poate fi „Unknown”.**
- Compilați codul și corectați evetualele erori.
- Rulați aplicația. În fereastra ce va apărea faceți click pe **Actions**, apoi click pe **Query**.
- Eticheta trebuie să afișeze un mesaj asemănător:
  Name: / Authn: False/Roles (none)
  Acest lucru demonstrează că nu există un principiu de thread autentificat și că identitatea nu are nume.
- Faceți click pe **Authenticate** și apoi pe **Exit** pentru a închide aplicația.

- Vizualizați codul și controalele clasei **Login.cs**. Duceți-vă la proprietatea **Principal**. Implementarea implicită pentru această proprietate este de a returna principiul de pe thread-ul curent.

Taskul 4: Testarea codului

- Compilați, rulați și testați aplicația:
  - Faceți click pe **Actions**.
  - Faceți click pe **Query**.
  - Faceți click pe **Authenticate**, apoi pe **Login**.
  - Datele din text box –uri nu sunt folosite, astfel faceți click pe **OK**.
  - Apare o căsuță de dialog care avertizează că utilizatorul nu este autentificat. Faceți click pe **OK**.
  - Apare fereastra **Order Form**.
  - Verificați că eticheta de jos a ferestrei **Query** indică faptul că principiul nu este autentificat.
- Realizați o interogare:
  - În căsuța **Product** (din fereastra Query), tastați ”Seat”.
  - Faceți click pe **Query**.
  - Verificați ca grupul **Results** să ofere aceleași valori precum tabelul de mai jos.

Etichetă	Valoare
Product	Seat
Stock	5
Price each	$29.99

- În fereastra **Order Form** realizați o comandă, urmând următoarele etape:
  - În căsuța **Product**, tastați ”Seat”.
  - Setați valoarea **Quantity** la 1.
  - Faceți click pe butonul **>>>.** Obiectul Seat este adăugat controlului listview.
  - Faceți click pe butonul **Order**.
  - Va apărea mesajul ”**Customer payment $29.99**”. Faceți click pe butonul **OK**.
  - În fereastra **Query** faceți click pe butonul Query pentru a interoga iar obiectul Seat. Verificați că s-a redus valoarea Stock la 4.
- Realizați o interogare:
  - În căsuța **Product**, tastați ”Wheel”.
  - Faceți click pe **Query**.
  - Verificați ca grupul **Results** să ofere aceleași valori precum tabelul de mai jos.

Etichetă	Valoare
Product	Wheel
Stock	10
Price each	$19.99

- În fereastra **Order Form** adăugați o roată comenzii și apoi cereți returnarea banilor:
  - În căsuța **Product**, tastați ”Wheel”.
  - Setați valoarea **Quantity** la 1.
  - Faceți click pe butonul **>>>.** Obiectul Wheel este adăugat controlului listview.
  - Faceți click pe butonul **Refund**.
  - Va apărea mesajul ”**Pay customer $19.99**”. Faceți click pe butonul **OK**.
  - În fereastra **Query** faceți click pe butonul **Query** pentru a interoga iar obiectul Seat. Verificați că a crescut valoarea **Stock** la 11.
- Faceți click pe Authenticate și apoi pe Exit pentru a închide aplicația.

➢ Exercițiul 2- Securitatea bazată pe roluri

În acest exercițiu veți adăuga codului securitate bazată pe roluri. Pentru aceasta veți schimba clasa Inventory, astfel încât doar principiile din rolurile specificate să poată accesa metodele care modifică inventarul. Dacă aceste metode sunt apelate de alte principii decât cele specificate se vor arunca niște excepții pe care va trebui să le tratați.

Taskul 1: Adăugarea unei clase personalizate principiale.

- Adăugați soluției un nou proiect de tipul **Class Library**, cu numele **Authentication**.
- Modificați namespace-ul implicit **Authentication** în **PointOfSale**.
- Pregătiți fișierul pentru codul principial personalizat. Ștergeți clasa **Class1** generată automat, apoi modificați fișierul astfel încât să puteți folosi tipurile din namespace-ul **System.Security.Principal**.
- În suportul de laborator se găsește un fișier **CustomPrincipal.txt**, care oferă clasa **CustomPrincipal**. Vizualizați codul pentru aceasta.
- Copiați acest cod în proiectul **Class Library**.
- Implementați o clasă personalizată **CustomIdentity**, care vă permite să determinați dacă un utilizator este autentificat sau nu. Clasa va implementa **IIdentity**.
  - Va avea două câmpuri **name**- pentru a reține numele de utilizator și un boolean **bAuthenticated** – pentru a specifica dacîă utilizatorul este sau nu autentificat.
  - Implementați constructorul care inițializează aceste câmpuri și proprietățile corespunzătoare interfeței.
- Implementați clasa User:
  - Clasa va avea următoarele câmpuri:

```
public  string Name;
public  string Password;
public  IList<string> Roles;
```

  - Implementați metoda
    public static User Create(string name, string password)
    care va returna o instanță a clasei User .
  - Implementați metoda
    `public void AddRole(string role).`

- Implementați metoda **AuthenticateUser**. Adăugați cod pentru această metodă astfel încât:
  - Vede dacă există un utilizator cu numele potrivit în colecția **users,** dacă utilizatorul nu este găsit, va fi returnat un principiu neautentificat cu username-ul.
  - Compară parola cu cea din colecția users, atunci va fi returnat un principiu cu o identitate autentificată, inițializat cu rolurile specificate în colecție.
- Adăugați cod în metoda IsInRole astfel încât:
  - Verificați că principiul are o colecție **roles**. Dacă nu are, atunci nu poate fi realizată o verificare de roluri.
  - Verifică dacă există rolul cerut în colecția **roles**.
- Compilați și corectați eventualele erori.

Taskul 2: Autentificarea credențialelor folosind clasa **Login**.

- În proiectul **PointOfSale**, veți folosi unitatea de asamblare **Authentication**, astfel trebuie să o adăugați ca referință proiectului.
- Vizualizați codul clasei **Login.cs**.
- Implementați proprietatea **Principal**, folosind metoda **AuthenticateUser** din clasa **CustomPrincipal**.

Taskul 3: Compilarea și testarea

- Compilați, rulați și testați aplicația:
  - Faceți click pe **Actions**.
  - Faceți click pe **Query**.
    Apare fereastra **Query**. Eticheta din josul ferestrei afișează:
    Name: / Authn: False / Roles: [none]

- Logați-vă ca un alt utilizator:
  - Faceți click pe butonul **Authenticate** și apoi pe **Login**.
  - Logați-vă cu Username = Tom și Password = tom.
  - Apare fereastra **Order Form**, care va avea titlul Tom.
  - Eticheta din josul ferestrei **Query** va afișa:
    Name: Tom / Authn: True / Roles: Clerk

- R epetați logarea pentru Username = Jane și Password = jane.
  - Eticheta din josul ferestrei **Query** va afișa:
    Name: Jane / Authn: True / Roles: Clerk Manager
- Faceți click pe butonul **Authenticate**, și apoi pe **Exit** pentru a închide aplicația.

Taskul 4: Verificări de permisiuni principiale

- Vizualizați fișierul **Inventory.cs** și adăugați cod astfel încât să puteți folosi tipurile din namespace-ul **System.Security.Permissions**.
- Adăugați metodei **Buy** un atribut **PrincipalPermissionAttribute**, deoarece poate fi apelaă doar de utilizatori cu rolul **Clerk**.
- Adăugați metodei **Refund** un atribut **PrincipalPermissionAttribute**, deoarece poate fi apelaă doar de utilizatori cu rolul **Manager**.
- Schimbați politica principială pentru domeniul de aplicație:
  - Duceți-vă în proiectul **PointOfSale**, în fișierul **Program.cs**.
  - În metoda **Main**, adugați cod pentru a seta politica principială.
  - Specificați principiul de thread implicit. Adăugați cod pentru a seta principiul de thread implicit la un obiect neautentificat **CustomPrincipal**.

```
AppDomain.CurrentDomain.SetPrincipalPolicy(PrincipalPolicy.WindowsPrincipal);
AppDomain.CurrentDomain.SetThreadPrincipal(CustomPrincipal.Unauthenticated);
```

- În fișierul **MainForm.cs**, implementați handler-ul **Logout** pentru a elibera principiul:
  - Duceți-vă la metoda **logoutToolStripMenuItem_Click**
  - La inceputul metodei adăugați cod pentru a adăuga principiul de thread la un principiu neautentificat.
- Compilați și corectați eventualele erori.

Taskul 5: Tratarea excepțiilor de securitate

- Deschideți fișierul **OrderForm.cs**
- Introduceți codul din metodele **btnOrder_Click** și **btnRefund_Click**, în blocuri try – catch, deoarece apelarea metodei **Inventory.Buy**, respectiv **Inventory.Refund** poate arunca excepții.
- Compilați și corectați eventualele erori.

➢ Exercițiul 3- Testarea aplicației

În acest exercițiu veți testa aplicația cu scopul de a investiga securitatea bazată pe roluri. Veți încerca să realizați cele trei acțiuni de interogare a inventarului, cumpărare a unui obiect și returnarea acestuia pentru fiecare dintre cele trei tipuri de utilizatori: utilizator neautentificat, utilizator cu rolul Clerk și utilizator cu rolul Manager.

Taskul 1: Testarea codului cu utilizator neautentificat.

- Compilați, rulați și testați aplicația:
  - Faceți click pe **Actions**.
  - Faceți click pe **Query**.
    Apare fereastra **Query**. Eticheta din josul ferestrei afișează:

Name: Unauthenticated / Authn: False / Roles: [none]
- Faceți click pe **Authenticate**, apoi pe **Login**.
- Datele din text box –uri nu sunt folosite, astfel faceți click e OK.
- Apare o căsuță de dialog care avertizează că utilizatorul nu este autentificat. Faceți click pe **OK**.
- Apare fereastra **Order Form**.
  Codul rulează sub un utilizator neautentificat.

- Interogați inventarul pentru obiectul Seat:
  - În căsuța **Product**, tastați ”Seat”.
  - Faceți click pe **Query**.
  - Verificați ca grupul **Results** să ofere aceleași valori precum tabelul de mai jos.

Etichetă	Valoare
Product	Seat
Stock	5
Price each	$29.99

Astfel se demonstrează că un utilizator neautentificat poate să interogheze inventarul.

- În fereastra **Order Form** realizați o comandă, urmând următoarele etape:
  - În căsuța **Product**, selectați ”Seat”.
  - Setați valoarea **Quatity** la 1.
  - Faceți click pe butonul **>>>.** Obiectul **Seat** este adăugat controlului listView.
  - Faceți click pe butonul **Order**.

  Se demonstrează că un utilizator neautentificat nu poate să realizeze o comandă.

  - În fereastra **Query** faceți click pe butonul **Query** pentru a interoga iar obiectul Seat. Verificați că valoarea **Stock** rămâne 5.
- Realizați o interogare:
  - În căsuța **Product**, tastați ”Wheel”.
  - Faceți click pe **Query**.
  - Verificați ca grupul **Results** să ofere aceleași valori precum tabelul de mai jos.

Etichetă	Valoare
Product	Wheel
Stock	10
Price each	$19.99

- În fereastra **Order Form**, adăugați o roată comenzii și apoi cereți returnarea banilor:
  - Faceți click pe butonul **New** pentru a elibera form-ul.
  - În căsuța Product, tastați ”Wheel”.
  - Setați valoarea **Quatity** la 1.
  - Faceți click pe butonul **>>>.** Obiectul **Wheel** este adăugat controlului listview.
  - Faceți click pe butonul **Refund**.

  Se demonstrează că un utilizator neautentificat nu poate să realizeze o returnare de plată.

- În fereastra **Query** faceți click pe butonul **Query** pentru a interoga iar obiectul **Wheel**. Verificați că valoarea **Stock** a rămas la 10.

Taskul 2: Testarea codului folosind rolul **Clerk**

- Logați-vă ca un alt utilizator:
  - Faceți click pe butonul **Authenticate** și apoi pe **Login**.
  - Logați-vă cu Username = Tom și Password = tom.
  - Apare fereastra **Order Form**, care va avea titlul Tom.
  - Faceți click pe **New** pentru a elibera formularul.

  Eticheta din josul ferestrei **Query** va afișa:
  Name: Tom / Authn: True / Roles: Clerk

- În fereastra **Order Form**, realizați o comandă:
  - În căsuța **Product**, tastați ”Seat”.
  - Setați valoarea **Quatity** la 1.
  - Faceți click pe butonul **>>>.** Obiectul **Seat** este adăugat controlului listview.
  - Faceți click pe butonul **Order**.

  Va apărea mesajul ”**Customer payment $29.99**”. Faceți click pe butonul OK. Astfel se demonstrează că un utilizator autentificat poate să realizeze o comandă.

  - În fereastra **Query** faceți click pe butonul **Query** pentru a interoga iar obiectul Seat. Verificați că valoarea **Stock** a scăzut la 4.

  Utilizatorul Tom are rolul **Clerk**, astfel poate să realizeze o comandă.

- Realizați o interogare:
  - În căsuța **Product**, tastați ”**Wheel**”.
  - Faceți click pe **Query**.
  - Verificați că valoarea este 10.
    Stock
- În fereastra **Order Form**, adăugați o roată comenzii și apoi cereți returnarea banilor:
  - În căsuța **Product**, tastați ”Wheel”.
  - Setați valoarea **Quatity** la 1.
  - Faceți click pe butonul **>>>.** Obiectul Wheel este adăugat controlului listView.
  - Faceți click pe butonul **Refund**.

  Se demonstrează că utilizatorii cu rolul **Clerk** nu pot să realizeze o returnare de plată.

  - În fereastra **Query** faceți click pe butonul **Query** pentru a interoga iar obiectul **Wheel**. Verificați că valoarea Stock a rămas la 10.

Taskul 3: Testarea codului folosind rolul **Manager**

- Logați-vă ca un alt utilizator **Manager**:
  - Faceți click pe butonul **Authenticate** și apoi pe **Login**.
  - Logați-vă cu **Username** = Jane și **Password** = jane.
  - Apare fereastra **Order Form**, care va avea titlul Jane

- Înc ercați să realizați o returnare de plată. Ce observati?