Lab 5 Serializacja

Mateusz Wiśniewski
Politechnika Śląska

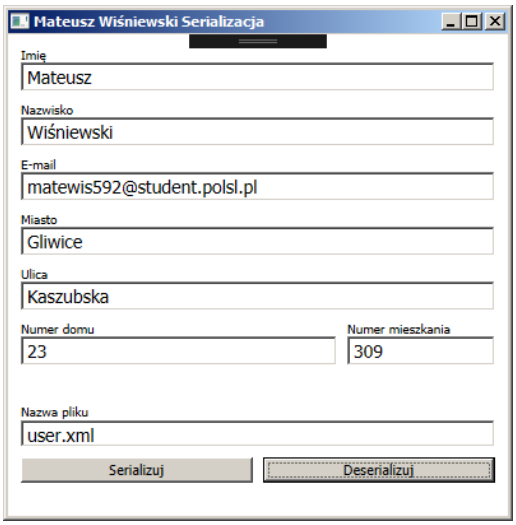
Zadanie polegało na zaprojektowaniu i stworzeniu aplikacji, która przechowuje dane klienta sklepu internetowego, ponad to wczytuje dane z pliku *.xml oraz dopisuje dane do pliku, natomiast gdy plik nie istnieje aplikacja go tworzy.

Aplikacja została stworzony za pomocą języka C# z wykorzystaniem technologii WPF oraz środowiska Microsoft Visual Studio 2017.

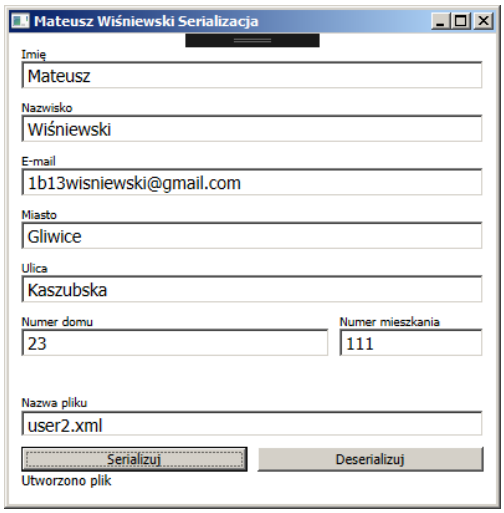
Plik user.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<User xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <FirstName>Mateusz</FirstName>
  <LastName>Wiśniewski</LastName>
  <Email>matewis592@student.polsl.pl</Email>
  <City>Gliwice</City>
  <Street>Kaszubska</Street>
  <HouseNumber>23</HouseNumber>
  <FlatNumber>309</FlatNumber>
</User>
```

Wczytane dane



Po zmianie danych oraz nazwie pliku aplikacja tworzy nowy plik o podanej nazwie oraz wyświetla komunikat „Utworzono plik”



Plik user2.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<User xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <FirstName>Mateusz</FirstName>
  <LastName>Wiśniewski</LastName>
  <Email>1b13wisniewski@gmail.com</Email>
  <City>Gliwice</City>
  <Street>Kaszubska</Street>
  <HouseNumber>23</HouseNumber>
  <FlatNumber>111</FlatNumber>
</User>
```

Kod serializacji

```
private void Serialize(object sender, RoutedEventArgs e)
{
    string firstName = tbox_first_name.Text.ToString();
    string lastName = tbox_last_name.Text.ToString();
    string email = tbox_email.Text.ToString();
    string city = tbox_city.Text.ToString();
    string street = tbox_street.Text.ToString();
    string houseNumber = tbox_house_number.Text.ToString();
    string flatNumber = tbox_flat_number.Text.ToString();
    User user = new User(firstName, lastName, email, city, street, houseNumber, flatNumber);

    string path = tbox_path.Text.ToString();

    if (!File.Exists(path))
    {
        File.Create(path).Close();
        tbox_message.Text = "Utworzono plik";
    }

    XmlSerializer serializer = new XmlSerializer(typeof(User));
    StreamWriter streamWriter = new StreamWriter(path);

    serializer.Serialize(streamWriter, user);

    streamWriter.Close();
}
```

Kod deserializacji

```
private void Deserialize(object sender, RoutedEventArgs e)
{
    string path = tbox_path.Text.ToString();

    XmlSerializer serializer = new XmlSerializer(typeof(User));

    try
    {
        StreamReader streamReader = new StreamReader(path);

        User user = (User)serializer.Deserialize(streamReader);
        streamReader.Close();

        tbox_first_name.Text = user.FirstName;
        tbox_last_name.Text = user.LastName;
        tbox_email.Text = user.Email;
        tbox_city.Text = user.City;
        tbox_street.Text = user.Street;
        tbox_house_number.Text = user.HouseNumber;
        tbox_flat_number.Text = user.FlatNumber;
    }
    catch (FileNotFoundException)
    {
        tbox_message.Text = "Brak pliku o podanej nazwie";
    }
    catch (Exception)
    {
        tbox_message.Text = "Błąd";
    }
}
```

Klasa User

```
public class User
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Email { get; set; }
    public string City { get; set; }
    public string Street { get; set; }
    public string HouseNumber { get; set; }
    public string FlatNumber { get; set; }
    public User() { }

    public User(string firstName, string lastName, string email, string city, string street, string houseNumber, string flatNumber)
    {
        this.FirstName = firstName;
        this.LastName = lastName;
        this.Email = email;
        this.City = city;
        this.Street = street;
        this.HouseNumber = houseNumber;
        this.FlatNumber = flatNumber;
    }

    public override string ToString()
    {
        return this.FirstName + ", " + this.LastName + ", " + this.Email;
    }
}
```

Jaki sens ma tego typu serializacja?

Główną zaletą tego typu serializacji jest prostota danych, XML posiada prosty zapis oraz czytelny plik, dzięki czemu „zwykły” człowiek również może w bardzo prosty i szybki sposób odczytać dane nawet bezpośrednio z pliku. Kolejną zaletą dzięki prostocie pliku XML jest przenoszenie między systemami jak i aplikacjami, ponieważ możemy plik XML przesłać na przykład na serwer i odczytywać go w aplikacji napisanej przy użyciu na przykład Javy.