

# Szerializáció

Tóth Zsolt

Miskolci Egyetem

2014

## 1 Szerializációs Alapfogalmak

## 2 Szerializációs Megoldások

- Object Szerializáció
- XML Szerializáció

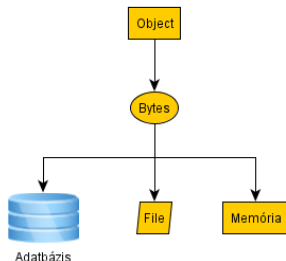
## Osztály

- Definíció
- Megadja:
  - ▶ Tulajdonságokat
  - ▶ Viselkedéseket
  - ▶ Kapcsolatokat
    - ★ Osztályokkal (Öröklődés, tartalmazás)
    - ★ Külvilággal (Interface)
- Egyetlen előfordulás
- Fájlrendszeren tárolódik

## Objektum

- Egyedi Entitás (oid)
- Típusa van (Osztály)
- Saját állapot
- Memóriát foglal
- Memóriában tárolódik
- Megszűnik, értéke elvész

- Aktuális állapot megőrzése
- Mentés
  - ▶ Aktuális állapot
  - ▶ Összesítő adatok
  - ▶ Biztonság
- Kommunikáció
  - ▶ Alkalmazások között
  - ▶ Hálózaton keresztül
- Adatbázist nem tárgyaljuk



## Mentés

- Saját adatformátum
  - ▶ Tömörebb (lehet)
  - ▶ Költséges
  - ▶ Módosítás?
- Adatbázis
  - ▶ Szerver
  - ▶ Szaktudás
  - ▶ Hálózat
  - ▶ SQLite

## Kommunikáció

- Saját fájlformátum
  - ▶ Hatékony
  - ▶ Költséges
  - ▶ Módosítás?
- Saját protokoll
  - ▶ Költséges
  - ▶ Bonyolult
  - ▶ Hibalehetőségek
  - ▶ Tűzfalak
  - ▶ stb.

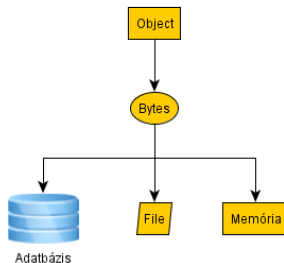
- Hatékony
- Általános
- Egyszerű
- .Net Framework része
- Számos formátum
  - ▶ Bináris
  - ▶ XML
  - ▶ SOAP

## Serializálható elemek

- Egyes objektumok
- Egyszerű adattagok
- Tömbök
- Kollekciónok

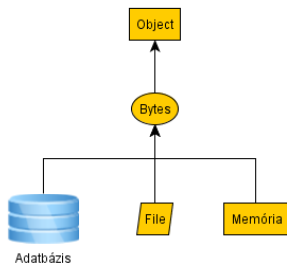
# Serializáció

- Eljárás
- Objektum -> Byte
- Objektum -> Text
- Adott API metódusok
- Egyszerű konverzió
- Fájlműveletek
- Megkötések
- [Serializable]



# Deszerializáció

- Eljárás
- Byte -> Objektum
- Text -> Objektum
- Adott API metódusok
- cast-olás





## 1 Szerializációs Alapfogalmak

## 2 Szerializációs Megoldások

- Object Szerializáció
- XML Szerializáció

# Objektum Szerializáció

- Object -> Byte
- Byte -> Object
- Számos formátum
  - ▶ Bináris
  - ▶ JSON
  - ▶ SOAP
- [Serializable]
- [NonSerialized]
- SerializationException

```
using System;  
  
[Serializable]  
class Person{  
    .  
    .  
    .  
}
```

- System.Runtime.Serialization
- Interface
- Serializált objektumok formázása.

```
void Serialize( Stream serializationStream, Object  
graph);
```

```
Object Deserialize( Stream serializationStream);
```

- System.Runtime.Serialization.Formatters.Binary
- Bináris formázás
- Object -> Byte
- Egyszerű
- [ Serializable ] attribútum

- Osztályok közötti hierarchia [Serializable]
- Kiterjesztés `class C1 {...}`
- Hatással van a szerializációra [Serializable]
- Őst is kell szerializálni `class C2 : C1 {...}`

# Példa

```
using System;

[Serializable]
class Person
{
    private string _name;
    private double _salary;

    public Person(string name, double salary) {...}
    public override string ToString() {...}
}
```

# Serialization Példa

```
Person p = new Person("John Doe", 100);
Stream outputStream = new
    FileStream(BinFile, FileMode.Create);
BinaryFormatter bf = new BinaryFormatter();
//Serializaiton
bf.Serialize(outputStream, p);
outputStream.Close();
//Deserializtaion
Stream inputStream = File.OpenRead(BinFile);
p = (Person) bf.Deserialize(inputStream);
outputStream.Close();
```

- 1 Szerializációs Alapfogalmak
- 2 Szerializációs Megoldások
  - Object Szerializáció
  - XML Szerializáció



- eXtensible Markup Language Elterjedt
- Jelölő nyelv (HTML)
- Szöveges állomány
- Adattárolás
- Hierarchikus adatstruktúra
- Gépi feldolgozásra
- XHTML
- Beállítások tárolása
- Android
- SOAP
- ...

- System.Xml.Serialization
  - XmlSerializer osztály
  - Nyilvános jelzőket igényel
  - Üres konstruktor
  - private adattag
  - public Property
- Nem kell [Serializable]
  - Számos attribútum:
    - ▶ XmlElement
    - ▶ XmlRoot
    - ▶ XmlArray
    - ▶ XmlArrayItem

- Osztály
- XML szerializálásért felel
- XmlSerializer(Type)
- typeof operátor
- Serialize(Stream, Object)
- Deserialize(Stream)

# Példa

```
public class Person
{
    private int _age;
    private string _name;
        public string Name { get; set; }
        public double Salary { get; set; }

    public Person() {}
    public Person(string name, double salary, int
        age) {...}
}
```

# Példa

```
Person p = new Person("John Doe", 100, 25);
XmlSerializer serializer = new
    XmlSerializer(typeof(Person));
var file = new FileInfo("person.xml");

//serialization
Stream outputStream = file.OpenWrite();
serializer.Serialize(outputStream, p);
outputStream.Close();

//deserialization
Stream inputStream = file.OpenRead();
p = (Person) serializer.Deserialize(inputStream);
inputStream.Close();
```

```
<?xml version="1.0"?>
<Person
xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd=
"http://www.w3.org/2001/XMLSchema">
  <Name>John Doe</Name>
  <Salary>100</Salary>
</Person>
```