

Увод у објектно програмирање:

Датотеке

Професор: **др Светлана Штрбац-Савић**

Маил / Кабинет: **svetlanas@viser.edu.rs / 501**

- Подаци се у току извршавања програма налазе у радној, оперативној меморији.
- Датотеке,
- Базе података,
- Регистар, и
- Интерне структуре.

Врсте датотека

- Секвенцијалне
- Директни приступ
- Бинарне датотеке

Примена датотека

- Подаци у апликацији морају ефикасно да користе диск (нпр., .mdb фајлови су огромни). Базе података по правилу заузимају знатно више меморијског простора него обичне секвенцијалне датотеке.
- Апликација се мора дистрибуирати на најефикаснији начин, на пример преко Интернета, пожељно је да датотеке буду мале и да се лако преузимају (базама требају DLL датотеке па ће величина датотеке бити превелика).

Примена датотека

- Подаци су већ у датотеци. Врло често су подаци које апликација треба да обради већ сачувани у облику датотеке. Рецимо, подаци се прикупљају преко неког аквизиционог система.
- Нека друга апликација треба да увезе или обради податке. Ако исте податке треба да обрађује нека друга апликација у централном (main frame) рачунару или на пример Excel, погодније је да подаци буду сачувани у облику секвенцијалне датотеке.
- Податке је потребно прегледати у едитору текста (дневник грешака).

FileSystem

- Класа **FileSystem** служи за давање информација и коришћење ресурса рачунара који се односе на партиције хард диска, директоријуме и различите типове датотека.
- За рад са датотекама користе се следећи методи класе **FileSystem**:
- **FileExists(*putanja*)** – Проверава да ли постоји датотека на наведеној путањи.
- **DeleteFile(*putanja*)** – Брише датотеку која је назначена у аргументу методе.
- **RenameFile(*putanja*, *novinaziv*)** – Мења назив датотеке у назив који је дат у другом аргументу методе.
- **GetFiles(*putanja*)** – Враћа листу свих датотека које се налазе на датој путањи.

FileSystem

- **CopyFile**(*staraPutanja, novaPutanja, pregazi*) – Копира датотеку назначену у првом параметру методе на локацију која је задата другим параметром. Трећи параметар може бити True или False . Он означава шта ће се догодити у колико на новој локацији већ постоји датотека са истим именом. Ако је вредност овог параметра True постојећа датотека биће обрисана, а на њено место биће копирана датотека која се копира.
- **MoveFile**(*staraPutanja, novaPutanja, pregazi*) Премешта датотеку назначену у првом параметру методе на локацију која је задата другим параметром. Трећи параметар може бити True или False . Он означава шта ће се догодити у колико на новој локацији већ постоји датотека са истим именом. Ако је вредност овог параметра True постојећа датотека биће обрисана, а на њено место биће пребачена датотека која се копира.

FileOpen функција

- Функција **FileOpen** служи за отварање датотеке
- **Public Sub FileOpen(_ ByVal FileNumber As Integer, _ ByVal FileName As String, _
ByVal Mode As OpenMode, _ Optional ByVal Access As OpenAccess = _
OpenAccess.Default, _ Optional ByVal Share As OpenShare = _
OpenShare.Default, _ Optional ByVal RecordLength As Integer = -1 _)**

FileOpen функција

- **FileNumber** - **Обавезан**. Сваки могући природан број и он представља број датотеке.
- **FileName** - **Обавезан**. **String** израз којим дефинишете име датотеке. Може да садржи име директоријума и диска.
- **Mode** - **Обавезан**. **Enum** израз којим дефинишете тип приступа: **Append**, **Binary**, **Input**, **Output** (за секвенцијални и бинарни приступ подацима) или **Random** (за директни приступ подацима у датотеци).
- **Access** – **Опциони аргумент**. Сваки могући природан број и он представља број датотеке. Користећи функцију **Read**, **Write** или **ReadWrite**. Подразумевана вредност је **ReadWrite**.

FileOpen функција

- **Share** - Опциони аргумент. . Enum израз којим се забрањују одређене операције над отвореном датотеком од стране других процеса: **Shared**, **Lock Read**, **Lock Write** и **Lock Read Write**. Подразумевана вредност је: **Lock Read Write**.
- **Recordlength** - Опциони аргумент. . Број мањи или једнак вредности 32,767 (bytes). За датотеке отворене у режиму случајног приступа (random access), ова вредност представља дужину записа. За датотеке са секвенцијалним приступом, *Recordlength* представља број карактера у међумеморији.

Mode	Opis
Append	Otvара datoteku sa ciljem sekvencijalnog dodavanja zapisa na njen kraj. Ako datoteka ne postoji Visual Basic je prvo kreira.
Binary	U Binary modu može se pristupiti podacima na nivou bajta. Znači mogu se čitati i upisivati pojedinačni bajtovi (znak po znak).
Input	Otvара datoteku za sekvencijalno čitanje od početka datoteke. Podacima se pristupa onim redosledom kojim su upisivani.
Output	Otvара datoteku za sekvencijalni upis startujući od početka datoteke. Ako datoteka već postoji i ima podatke oni će biti prebrisani (Overwrite)
Random	Otvара datoteku i za čitanje i za upis sa slučajnim ili direktnim pristupom.

FileOpen функција

- Ако је датотека већ отворена од стране неког другог процеса и одређени тип приступа није дозвољен, **FileOpen** операција ће изазвати грешку у извршавању програма.
- Ако датотека дефинисана аргументом **FileName** не постоји, она ће бити аутоматски креирана након отварања датотеке помоћу функције **FileOpen** за **Append**, **Binary**, **Output**, или **Random** приступ. Након што је датотека креирана, она се може референцирати користећи аргумент број датотеке `filename`.
- **Важно упозорење:** Ако датотека под наведеним именом већ постоји, отварањем датотеке у режиму `Output`, уништава се претходни садржај датотеке без икаквог упозорења од стране система. Дакле, биће уништени сви претходно унети подаци.

Функције

Namespace: Microsoft.VisualBasic

- **PrintLine (Print),**
- **WriteLine (Write),**
- **Input, LineInput, InputString**
- **EOF(int)** – враћа тачну вредност ако је крај датотеке

Употреба WriteLine функције

```
Private Sub Button1_Click(ByVal eventSender As System.Object,  
—  
ByVal eventArgs As System.EventArgs) Handles Button1.Click  
    Try  
        FileOpen(1, "c:/file.txt", OpenMode.Output)  
        WriteLine(1, Val(Text1.Text), Val(Text2.Text))  
        FileClose(1)  
    Catch ex As Exception  
        MsgBox("Greška!")  
    End Try  
End Sub
```

Употреба PrintLine функције

```
Private Sub Button1_Click(ByVal eventSender As System.Object,  
—  
ByVal eventArgs As System.EventArgs) Handles Button1.Click  
    Try  
        FileOpen(1, "c:/file.txt", OpenMode.Output)  
        PrintLine(1, TextBox1.Text)  
        FileClose(1)  
    Catch ex As Exception  
        MsgBox("Greška!")  
    End Try  
End Sub
```

Употреба Input функције

- **Input** функција се може користити за учитавање текстуалних или нумеричких података из секвенцијалних датотека.
- Када се датотека отвара за **Input**, датотека већ мора бити креирана иначе ће доћи до појаве грешке.
- У случајевима да се покуша отварање непостојеће датотеке са **Output** или **Append**, функција **FileOpen** креираће прво датотеку и затим ће је отворити.

Употреба Input функције

Input(FileNumber,Value)

```
Dim intFileNumIn As Integer
Dim InputFile As String
Dim intCtr As Integer
Dim intData As Integer
InputFile = "c:\PrintDemo.Txt"
intFileNumIn = 1
FileOpen(intFileNumIn, InputFile, FileMode.Input)
For intCtr = 1 to 10
    Input ( intFileNumIn, intData)
    MsgBox("Pročitani broj je: " + STR(intData))
Next intCtr
FileClose (intFileNumIn)
```

Употреба LineInput функције

```
Dim red As String
Try
    FileOpen(1,"c:/file.txt",FileMode.Input)
    Do Until EOF(1)
        red=LineInput(1)
        TextBox1.Text+=red+VbNewLine
    Loop
    FileClose(1)
Catch ex As Exception
    MsgBox("Greska")
End Try
```

StreamReader

- Класа **StreamReader** омогућава читање садржаја датотека.
- **StreamReader** класа се налази у System.IO Namespace-у и наследник је класе **TextReader**.
- Отварање датотеке за читање се остварује креирањем **StreamReader** објекта, при чему се путања датотеке и њено име наводе приликом креације објекта.
- Пример:

```
Dim citac As System.IO.StreamReader= New System.IO.StreamReader(txtBox1.Text)
```

Конструктори класе `StreamReader`

- Класа `StreamReader` има више конструктора:
- `StreamReader(putanja)` – иницијализује објекат `StreamReader` класе за име датотеке које параметар конструктора. *Putanja* је типа `String`.
- `StreamReader(putanja, encoding)` – иницијализује објекат `StreamReader` класе за име датотеке које параметар конструктора са назначеним енкодингом (`ASCII`, `UTF8`).
Putanja је типа `String`.
- Подразумевани енкодинг је `UTF8`.
- `Encoding.Unicode`, `Encoding.ASCII`, `Encoding.UTF32`, `Encoding.UTF7`,
`Encoding.UTF8`

Читање података из датотеке

- Читање података се из текстуалне датотеке може се извршити коришћењем метода `ReadAllText(putanja)` класе `FileSystem`.

Пример:

```
My.Computer.FileSystem.ReadAllText (putanja)
```

Секвенцијално читање података из датотеке

- Читање једног карактера текста из текстуалне датотеке може се извршити коришћењем методе **Read(putanja)** класе `SreamReader`.
- Ова метода чита караткер, по караткер датотеке док не прочита за крај датотеке и помера текућу позицију за наредно чинатање на карактер који следи.
- Да би прочитали целу датотеке у каратер по карактер, ову методу треба користити у комбинацији са неком од наредби понављања (петљи).
- Метода **ReadToEnd(putanja)** класе `SreamReader`, чита садржај датотеке од текуће позиције до краја датотеке.

Секвенцијално читање података из датотеке

- Читање једног реда текста из текстуалне датотеке може се извршити коришћењем методе `ReadLine(putanja)` класе `SreamReader`.
- Ова метода чита ред, по ред датотеке и помера текућу позицију за наредно чинатање на ред текста који следи.
- Да би прочитали целу датотеку у ред по ред, ову методу треба користити у комбинацији са неком од наредби понављања (петљи)

Пример:

```
Dim citac As System.IO.StreamReader= New  
System.IO.StreamReader(txtBox1.Text)  
Dim liniја As String  
...  
While True  
    liniја=citac.ReadLine()  
    If IsNothing(liniја) Then Exit While  
End If  
    If liniја.Length = 0 Then Continue While  
End If  
    txtPrikaz.Text&=liniја & vbCrLf  
End While  
Citac. Close()
```


Затварање датотеке

- Датотеку која је отворена за читање или писање потребно је затворити да би други корисници могли да је неометано користе.
- У претходном примеру коришћена је метода `Close()`, која ослобађа све ресурсе који су везани за `StreamReader` објекат.
- Ова метода нема параметре. Примењује се над конкретним објектом, у нашем примеру објектом `citac.Close()`, па је јасно која ће датотека бити затворена.

Упис података у датотеку

- Упис података се у текстуалне датотеке може се извршити коришћењем метода `WriteAllText(putanja, tekst, append)` класе `FileSystem`.
- Аргумент `append` је логичког типа. Ако му је вредност **True** додаће садржај *tekst* на крај претходног садржаја, а у супротном ће преко претходног текста датотеке уписати садржај из параметра *tekst*.

Пример:

```
My.Computer.FileSystem.WriteAllText (putanja,  
txtPrikaz.Tekst, False)
```

Секвенцијални упис података у датотеку

- Уписивање једног карактера текста у текстуалну датотеку може се извршити коришћењем методе **Write**(podatak) класе StreamWriter.
- Параметар података може бити разних типова: String, Object, UInt32, UInt64, Single... Ако је у питању било који тип податка који је нумерички или било који који није типа String, он се прво трансформише у String, па се уписује.

Секвенцијални упис података у датотеку

- Уписивање једног реда текста у текстуалну датотеку може се извршити коришћењем методе `WriteLine(podatak)` класе `SreamReader`.
- У колико се садржај уписује у датотеку која не постоји, прво ће датотека са именом из путање бити креирана, а затим ће садржај бити уписан.
- Метода `WriteLine()` уписује у датотеку ознаку новог реда.

Пример:

...

```
Dim pisac As System.IO.StreamWriter= New  
System.IO.StreamWriter(txtBox1.Text)  
Dim imena() As String={"Jelena","Biljana", "Svetlana",  
"Dusan"}  
For Each ime As String In imena  
    pisac.WriteLine(ime)  
Next  
pisac.Close()
```

Изузеци

- У циљу обраде нежељених ситуација користе се изузеци.
- До изузетака може доћи из различитих разлога:
 1. Недоступност ресурса (погрешна путања, уређај није спреман, екстерна меморија није доступна...)
 2. Недозвољен приступ ресурсу
 3. Прекорачење дозвољеног опсега
 4. Дељење нулом

Изузеци

- Кључне речи **Try** и **Catch** су део механизма изузетака (*exceptions, exception handling*).
- У **Try** блоку, дакле између **Try** и **Catch**, стављају се наредбе при чијем извођењу може доћи до неког изузетног стања (нпр. Грешка при упису у датотеку, недостатак папира у штампачу, прекид везе са мрежом...).
- Када се реализује изузетак, његов тип се упоређује са типом који стоји иза **Catch ... As** (*Exception* одговара свим изузецима). Ако је тип одговарајући, прелази се на обраду ухваћеног изузетка која најчешће подразумева уписивање описа грешке у неки .log фајл и/или обавештење корисника о грешци.

```
...  
Dim x As Integer;  
Dim y As Integer;  
try  
    y = 100/x  
catch (ArithmeticException e)  
    Console.WriteLine("ArithmeticException Handler: {0}", e.ToString())  
catch (Exception e) { Console.WriteLine("Generic Exception Handler:  
{0}", e.ToString())  
End Try
```



```
...  
try  
    Dim sr As System.IO.StreamWriter= New System.IO.StreamWriter(txtBox1.Text)  
  
    Dim line As String  
    while (not IsNothing(linija) )  
        Console.WriteLine(line)  
    catch (Exception e)  
        Console.WriteLine("The file could not be read:")  
        Console.WriteLine(e.Message)  
  
End Try
```

```
Dim Pokusaj As Short = 0
Try
    FileOpen(1, "a:/file.txt", OpenMode.Output)
    PrintLine(1, TextBox1.Text)
    FileClose(1)
    ' Bilo koja greška (izuzetak) koja se pojavila u Try bloku
Catch ex As Exception
    ' Ovde se samo javlja korisniku da je došlo do greške
    If Pokusaj <= 2 Then
        MsgBox("Ubacite disketu u disketnu jedinicu!")
    Else
        MsgBox("Ucitavanje datoteke sa diskete nije uspelo!")
    End If
End Try
```

```
Dim Pokusaj As Short = 0
Try
    FileOpen(1, "a:/file.txt", OpenMode.Output)
    PrintLine(1, TextBox1.Text)
    FileClose(1)
    ' Bilo koja greška (izuzetak) koja se pojavila u Try bloku
Catch ex As Exception
    ' Ovde se samo javlja korisniku da je došlo do greške
    If Pokusaj <= 2 Then
        MsgBox("Ubacite disketu u disketnu jedinicu!")
    Else
        MsgBox("Ucitavanje datoteke sa diskete nije uspelo!")
    End If
End Try
```