

C# Advanced – Class5

Trainer: Miodrag Cekikj – cekicmiodrag@gmail.com

Assistant: Andrej Chichakovski – andrejchichak@gmail.com



SEAVUS EDUCATION *and*
DEVELOPMENT CENTER

Code academy @ Skopje, 2019



Agenda



Delegates



Events





Delegates

- **delegate** keyword
- Delegate is a **pointer** to function(s).
- Delegate class – base class for all delegate types
- MulticastDelegate class
- Only the **system** and **compilers** can derive explicitly from the **Delegate class** or from the **MulticastDelegate class**.
- **Delegate is an object that knows how to call a method or group of methods.**
- BUT, we can call methods directly, WHY delegates, WHY pointers now? ☹



Delegates

- **Practical implementation:**
 - Designing extensible and flexible applications (for ex. **frameworks**)
- BUT, we have learned about interfaces, WHY this now? ☹️
- Delegates in case of:
 - **Event driven development** (Event design pattern needed).
 - **No need to access additional properties and methods** from the object implementing the logic (method).
 - **Personal preference** and way of thinking about some problem.
 - Very powerful concept in combination with Events.



Delegates

- Definition: **Delegate is a type-safe function pointer**: it points to a function and when a delegate is called that function will be used.
- Delegates offers flexibility:
 - **They have similar structure as functions**. The signature of the delegate must match the signature of the function it calls, otherwise you get a compiling error.
 - **Delegate is similar to class** in the way that you can make an instance of it and then call functions.
 - **Multi-cast delegate**: is a delegate that has **references to multiple functions**. In this case a **single call of the delegate calls each of the added functions to the delegate**.
 - * Declaration: A delegate can be declared “**anywhere**”: outside the namespace, inside the namespace, inside a class or nested class. It's scope/visibility depends on the place where it's defined.

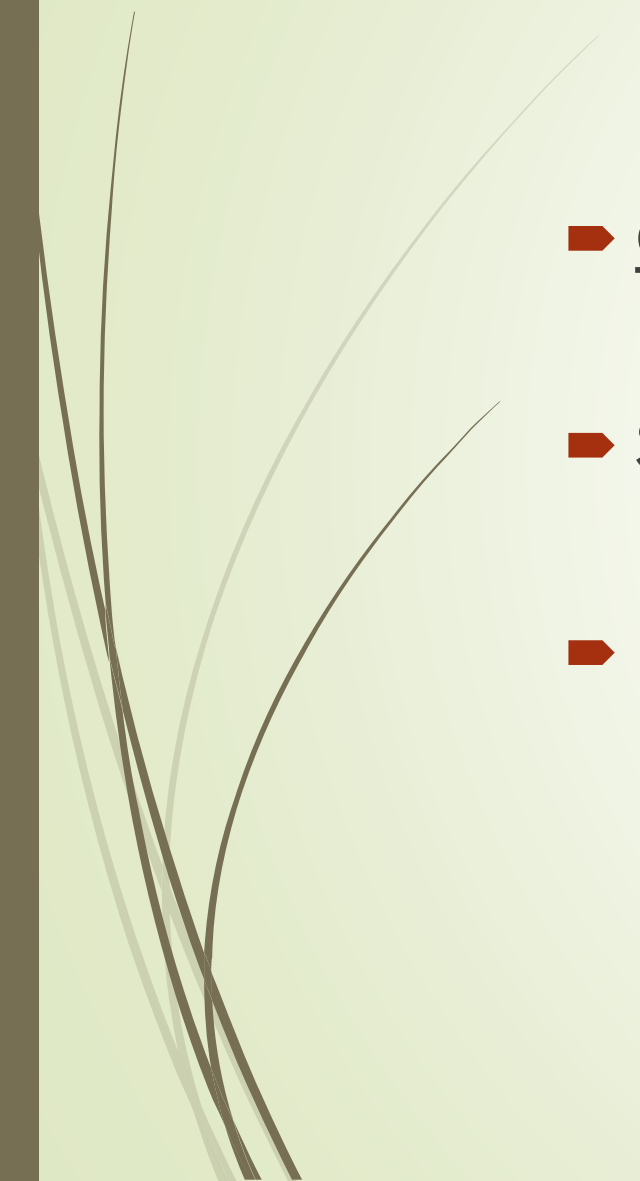


Delegates

- Demo
 - Video Editor Application



Events

- Communication between **classes** and **objects**.
 - Supports the **principle of extending the applications**.
 - Helps in building **Loosely Coupled Applications**.
- 



Events

- Definition: **Events are variables of type delegates.**
- Events enable a class or object to **notify** other classes or objects when something of interest occurs. The class that sends (or *raises*) the event is called the **PUBLISHER** and the classes that receive (or *handle*) the event are called **SUBSCRIBER**.

Events

➤ Demo

- Video Encoder application (with Event design pattern)

