C# and Microsoft .NET

Inheritance and Interfaces

Trainer: Georgi Panayotov
E-mail: smg@smg-bg.net
Phone / Viber: +359877347912

PRAGMATIC
www.pragmatic.bg

Last time…

- Was it fun?
- Was it helpful?
- A few slides from last^2 time ☺
  - Static
  - Namespaces
  - Class Libraries
  - Project References
  - NuGet Packages

Static

- What does the **static** keyword stand for?
- Static Fields, Constants and Properties
- Static Methods
- Static Constructors
- Static Classes
- Standard static classes in .NET
  - System.Console
  - System.Environment
  - System.Math

Namespaces

- What is a namespace
  - The **namespace** keyword is used to declare a scope that contains a set of related types
  - You can use a **namespace** to organize code elements and to create globally unique types
  - Used for logical organization of the source code
- Syntax
  - namespace namespace_name {}
  - namespace namespace_name.nested_namespace {}
- Default namespace for project
- Using **namespaces** (e.g. using keyword)

Class Libraries

- What is a Class Library
  - Defines namespaces with types (classes, structs, interfaces*, etc.) that can be reused across applications
  - Used for physical organization of the source code
- Project Templates
  - .NET Framework
  - .NET Core
  - .NET Standard
- Using Class Libraries / Project References
- What about NuGet ☺

Principles of OOP

- Abstraction

  Hide all but relevant details from end users / programmers
- Encapsulation

  Hide the internal details from end users / programmers
- Inheritance

  Reuse (share) logic and state from parent classes
- *Polymorphism**

Inheritance

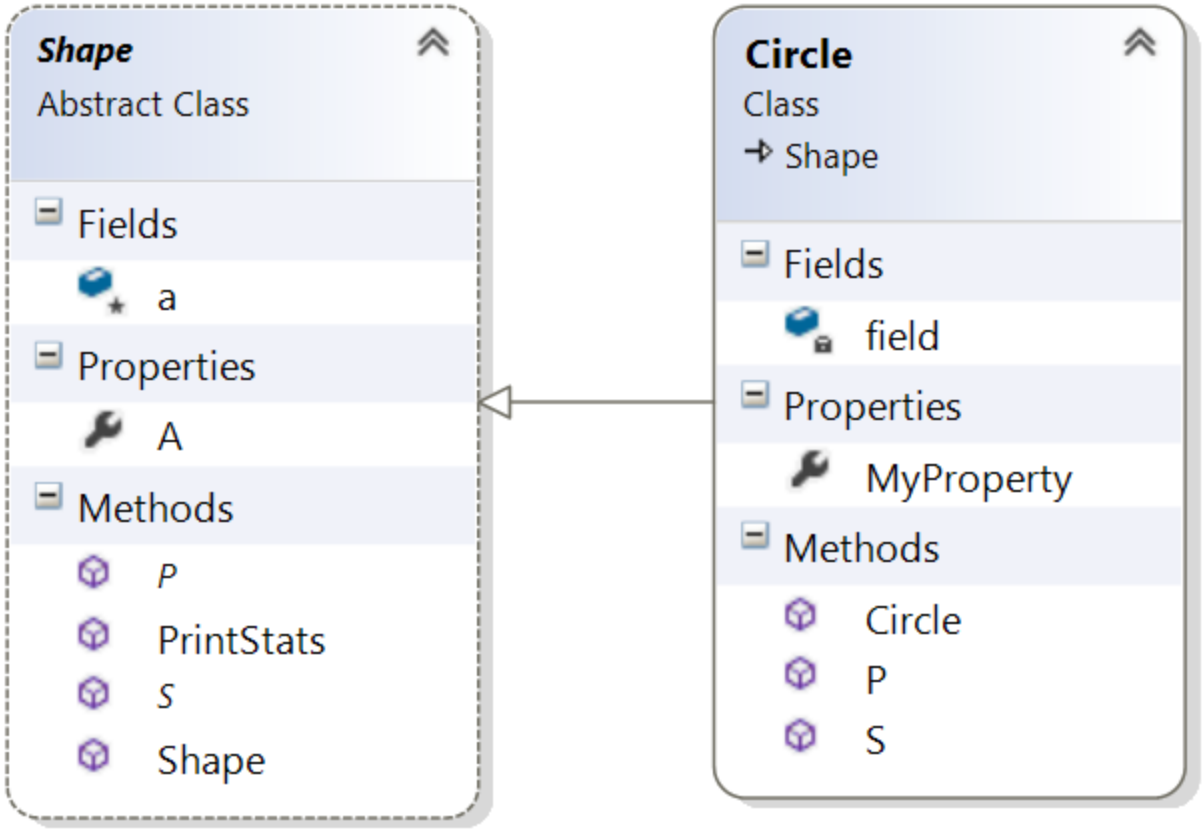
- The hierarchical structure of classes
- What exactly is inheritance
  - Derived classes inherit base classes
  - Classes implementing interfaces
  - Derived interfaces implementing base interfaces
- Uses of inheritance
- What about structs?

Inheritance

```
access_modifier class class_name : parent_class
{

}
```

Class Diagrams



The base class

- Accessing methods of the base
- Access_modifiers with inheritance
  - Protected
  - Internal protected
  - Private protected

The Ultimate Base Class

- Object
  - Object() **/* constructor */**
  - bool Equals(Object objA, Object objB) **/* static */**
  - bool ReferenceEquals(Object objA, Object objB) **/* static */**
  - bool Equals(Object obj)
  - int GetHashCode()
  - Type GetType()
  - string ToString()
  - MemberwiseClone(); **/* protected */**

Abstract classes

- Abstract classes
  - access_modifier abstract class class_name { … }
- Why placing abstractions
- Abstract classes and class hierarchies

Interfaces

- Interfaces vs. Abstract classes
- The concept of interfaces
- When do we use interfaces?
- Inheritance vs. Implementation
- Interfaces as contracts for client code
- Interfaces and design patterns


Questions?