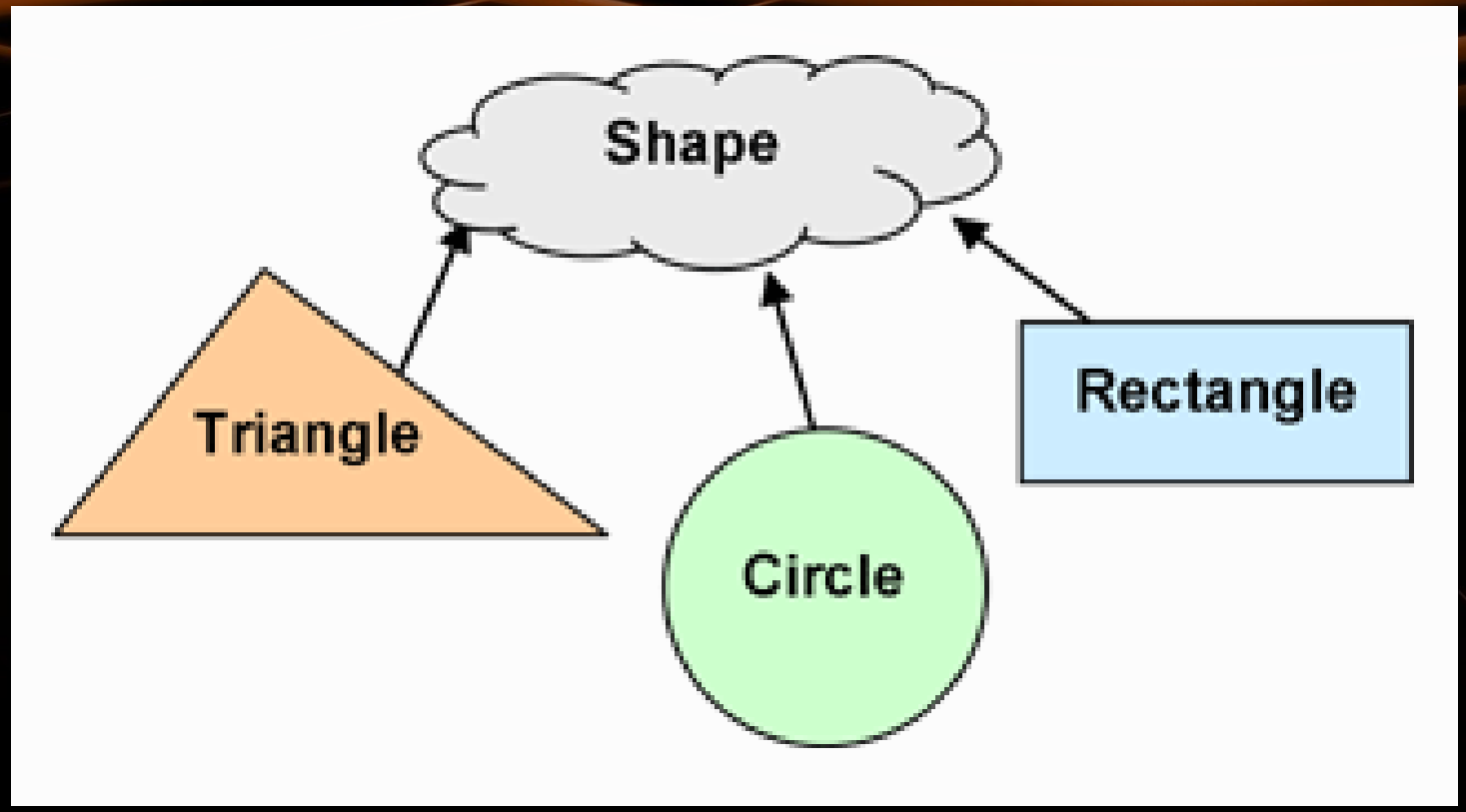
Абстрактни класове и полиморфизъм

Учителски екип

Обучение за ИТ кариера

https://it-kariera.mon.bg/e-learning/



Абстрактни класове

Абстрактни класове

- Абстрактните класове **НЕ МОГАТ** да бъдат инстанцирани

```
public abstract class Shape {}
public class Circle : Shape {}

Shape shape = new Shape(); // Грешка при компилиране
Shape circle = new Circle(); // полиморфизъм
```

- Абстрактният клас може да включва абстрактни методи, а може и да не включва такива.
- Ако клас има поне един абстрактен метод, той трябва да бъде деклариран като абстрактен
- За да използвате абстрактен клас, трябва да го наследите

Елементи на абстрактния клас

```
Public abstract class Shape
{
  private Point startPoint;
  protected Shape(Point startPoint) {
    this.startPoint = startPoint;
  }
  public StartPoint { return this.startPoint; }
  public abstract void Draw();
}
```

Може да има полета

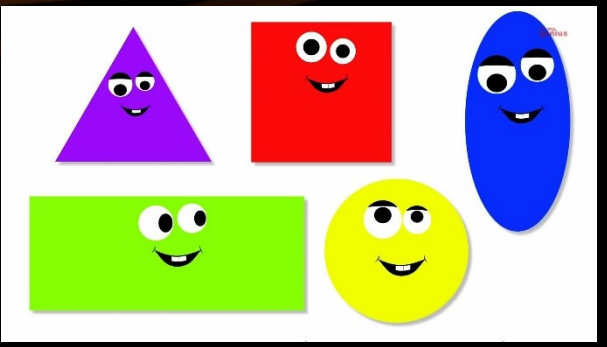
Може да има конструктор

Всеки абстрактен метод ТРЯБВА да се имплементира от подкласовете

Може да има методи с код в тях

Задача: Фигури

Shape
-double perimeter -double area
+CalculatePerimeter *+CalculateArea*



Rectangle
-double height -double width
+CalculatePerimeter +CalculateArea

Circle
-double height -double width
+CalculatePerimeter +CalculateArea

Задача: Фигури

```
public abstract class Shape
{
    public abstract double CalculatePerimeter();

    public abstract double CalculateArea();

    public virtual string Draw()
    {
        return "Drawing ";
    }
}
```

Задача: Фигури (2)

```
public class Rectangle : Shape
{
  //TODO: Добавете полета и конструктор
  public override double CalculatePerimeter()
  { return this.sideA * 2 + this.sideB * 2; }

  public override double CalculateArea()

  { return this.sideA * this.sideB; }

  public sealed override string Draw()

  { return base.Draw() + "Rectangle"; }
}
```

Solution: Shapes (3)

```
public class Circle : Shape
{
  //TODO: Добавете полета и конструктор
  public override double CalculatePerimeter()
  { return 2 * Math.PI * this.radius; }

  public override double CalculateArea()

  { return Math.PI * this.radius * this.radius; }

  public sealed override string Draw()

  { return base.Draw() + "Circle"; }
}
```

Ключова дума – Sealed

- Модификатора предотвратява други класове да наследяват съответния клас

```
public abstract class Shape {}
public sealed class Rectangle : Shape {}
public class Sqaure : Rectangle {}
//Грешка при компилиране
```

Ключова дума – Sealed

- Позволява наследяване от класа и предотвратява презаписване на конкретен виртуален метод или свойства.

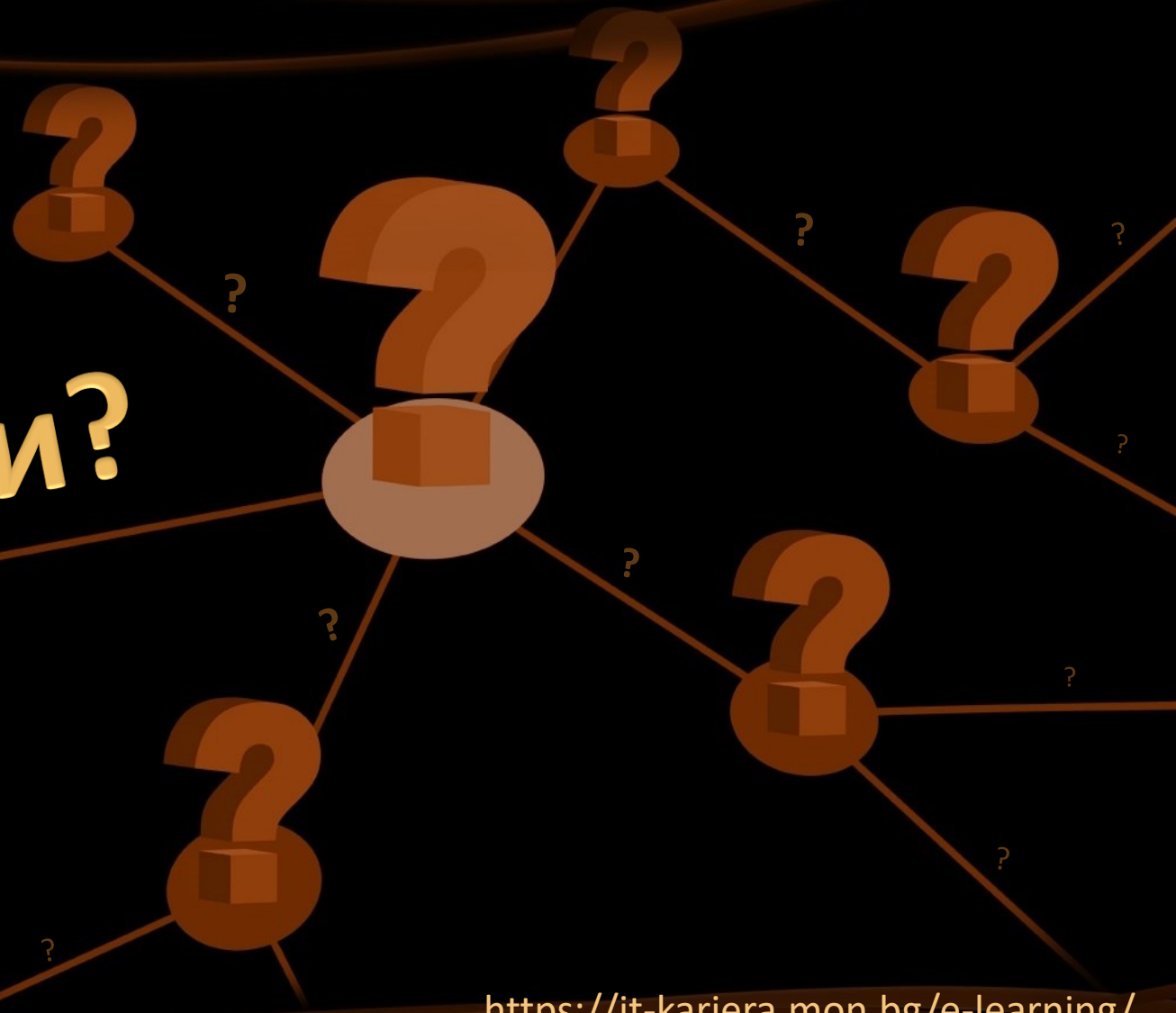
```
public class Rectangle : Shape
{
  public sealed override double GetArea() {}
}
public class Sqaure : Rectangle
{
  public override double GetArea() {}
  // Compile time error
}
```

Какво научихме днес?

- Абстрактни класове
- Абстрактни методи
- Употреба на sealed

Абстрактни класове и полиморфизъм

Въпроси?



https://it-kariera.mon.bg/e-learning/

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**