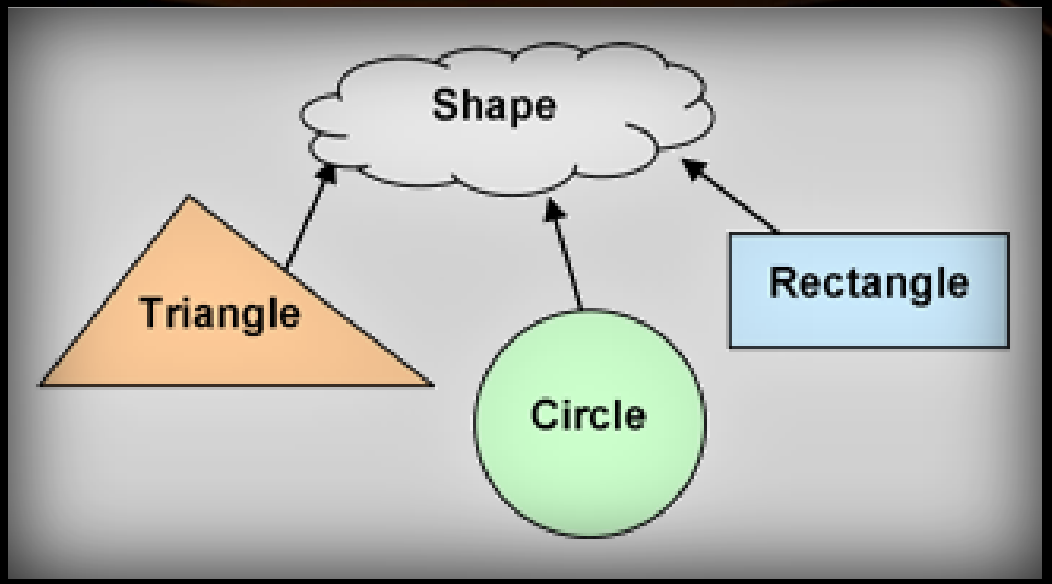
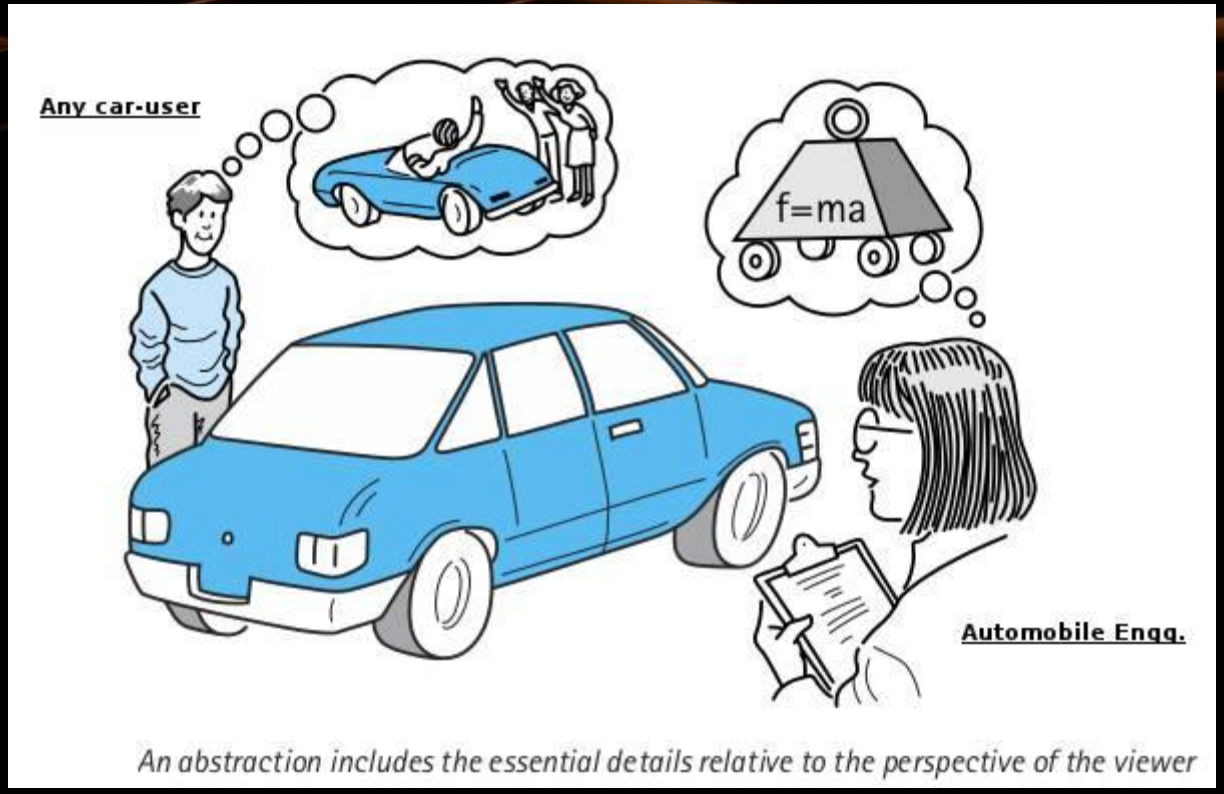
Абстрактни класове

Учителски екип

Обучение за ИТ кариера

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

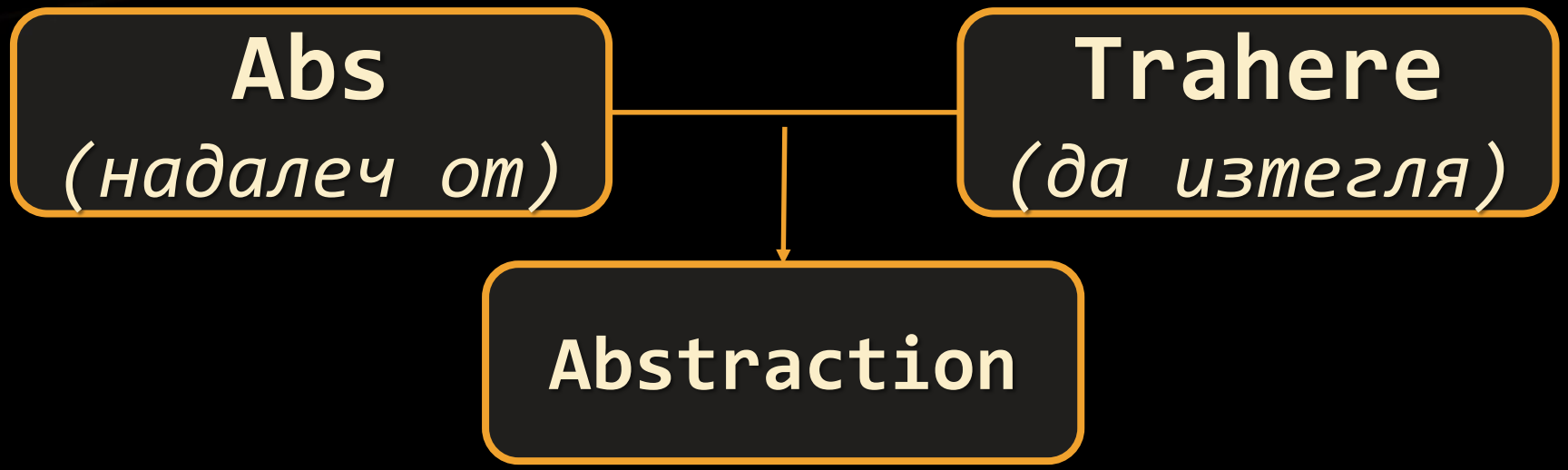



Any car-user
f=ma
Automobile Engg.
An abstraction includes the essential details relative to the perspective of the viewer

Абстракция

Какво е абстракция?

- От латински



Процес на отдалечаване или премахване на характеристики от нещо с цел да се намали до множество от основни характеристики.

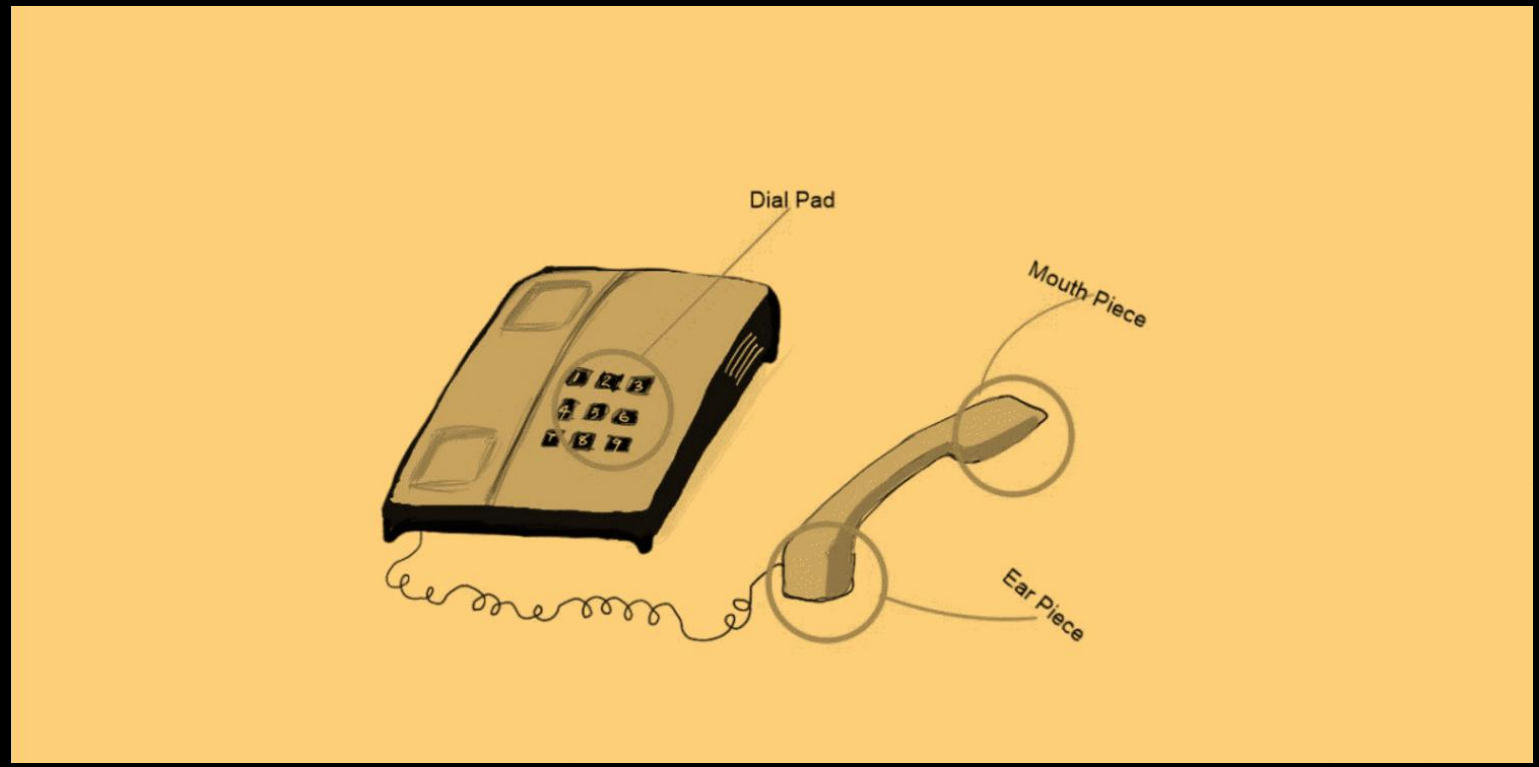
Абастракция в ООП

- Абстракцията означава да се ингорират несъществените черти, свойства или функции и да се наблегне на съществените...

- ... Съществени за проекта, по който работим
- Абстракцията помага да се управлява сложността

Пример за абстракция

- Абстракцията позволява да се фокусираме на това какво прави обекта вместо на това как го прави.



Как постигаме абстракция?

- Има два начина за постигане на абстракция
  - Интерфейси (100% абстракция)
  - Абстрактен клас (0% - 100% абстракция)

```
public interface IAnimal {}
public abstract class Mammal {}
public class Person : Mammal, IAnimal {}
```

Абстракция с/у Капсулация

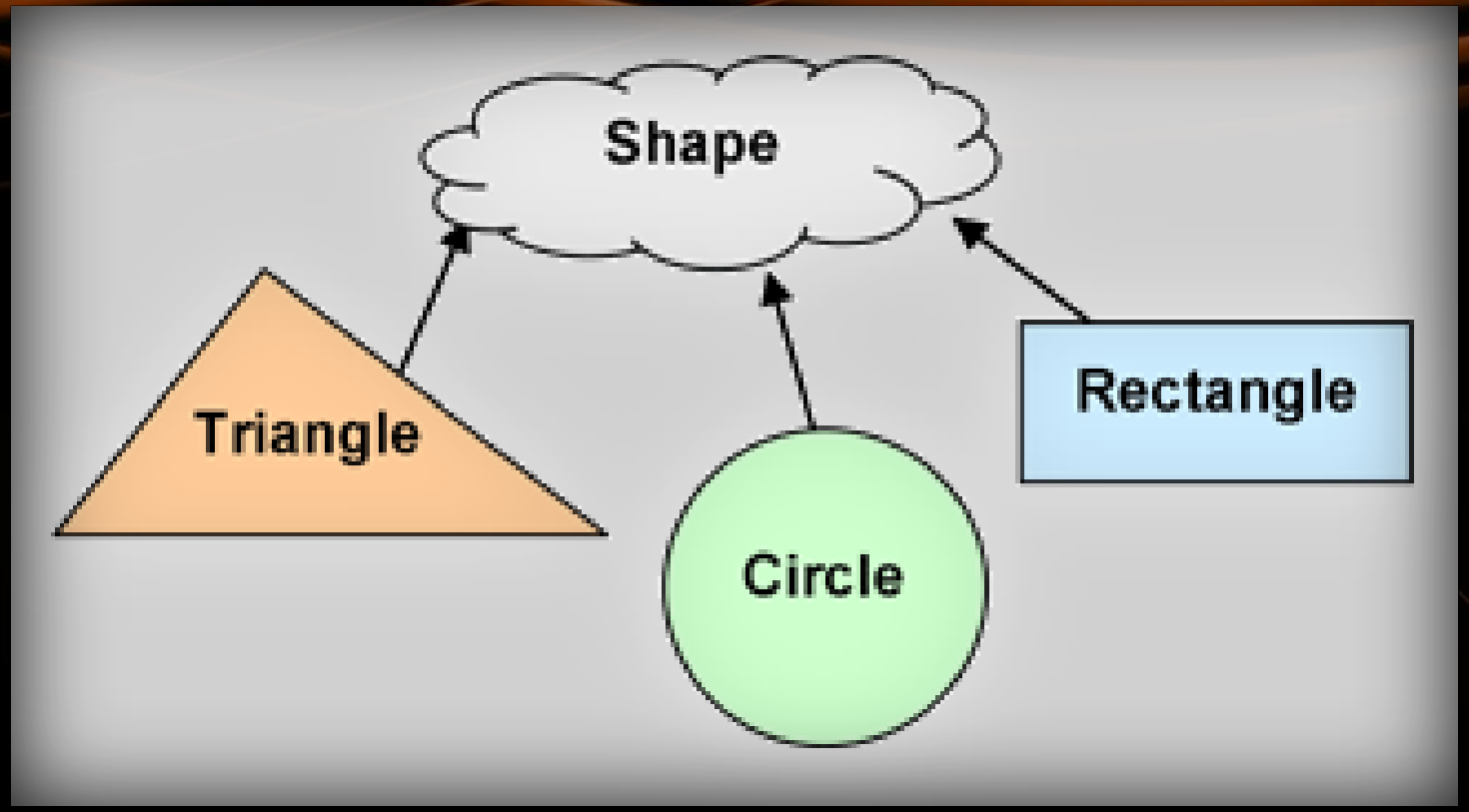
Абстракция

- Постига се чрез интерфейси и абстрактни класове
- Скрива подробностите по реализацията и показва само функционалностите към потребителя.

Капсулация

- Получава се чрез модификаторите за достъп (private, public…)
- Скрива кода и информацията в един компонент, за да я защити от външния свят

Абстракция с/у Капсулация (2)



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

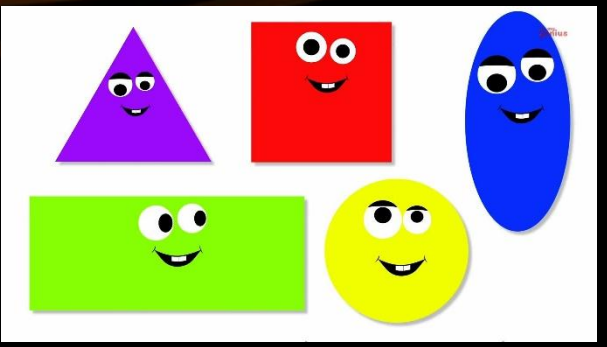
Може да има конструктор

Може да има методи с код в тях

Всеки абстрактен метод ТРЯБВА да се имплементира от подкласовете

Задача: Фигури

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



Rectangle
-double height -double width
+CalculatePerimeter +CalculateArea

Circle
-double radius
+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"; }
}
```

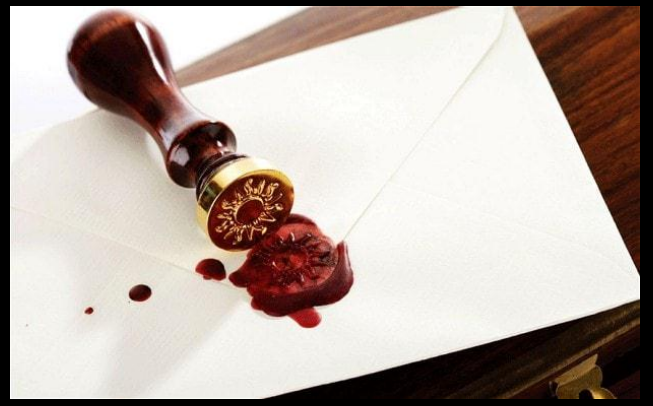
Solution: Shapes (3)

```
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.

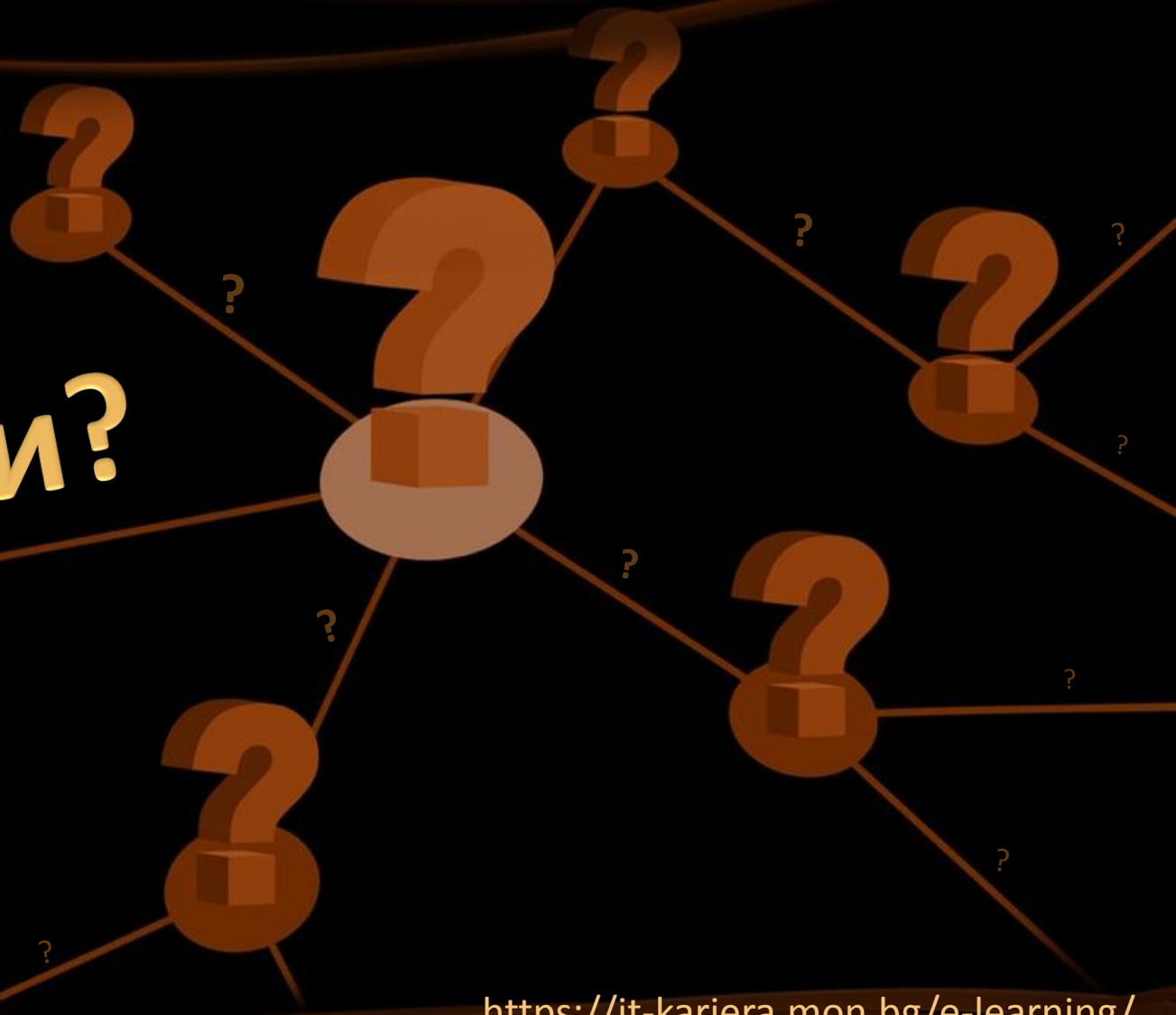
```
public class Rectangle : Shape
{
  public sealed override double GetArea() {}
}
public class Sqaure : Rectangle
{
  public override double GetArea() {}
  //Compile time error
}
```

Какво научихме днес?

- Абстрактни класове
- Абстрактни методи
- Употреба на sealed

Абстрактни класове

Въпроси?



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

Лиценз

- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"

- Благодарности: настоящият материал може да съдържа части от следните източници
  - Книга "Основи на програмирането със C#" от Светлин Наков и колектив с лиценз CC-BY-SA