**Доклад**

**Обектно ориентирано програмиране (*ООП*)**

Ваня 11а

1. Определение
2. Клас. Дефиниране на класове
3. Полета и своиства
4. Конструктор
5. Методи
6. Статични методи и конструктори
7. Статични полета и своиства
8. Източници

**1.Обектно ориентираното програмиране (*ООП*)**е фундаментален стил на програмиране, при която една програмна система се моделира като набор от обекти, които взаимодействат помежду си, за разлика от традиционното виждане, в което една програма е списък от инструкции, които компютърът изпълнява. Всеки обект е способен да получава съобщения, обработва данни и праща съобщения на други обекти.

**2. Клас. Дефиниране на класове**

Клас (class) в ООП наричаме описание (спецификация) на даден клас обекти от реал­ността. Класът представлява шаблон, който описва видо­вете състояния и поведението на конкретните обекти (екземплярите), които биват създа­вани от този клас (шаблон).

Обект (object) наричаме екземпляр създаден по дефиницията (описание­то) на даден клас. Когато един обект е създаден по описанието, което един клас дефинира, казваме, че обектът е от тип "името на този клас".

Всеки клас съдържа дефиниция на това какви данни трябва да се съдържат в един обект, за да се опише състоянието му. Обектът (конкретния екземпляр от този клас) съдържа самите данни. Тези данни дефинират състоянието му.

Освен състоянието, в класа също се описва и поведението на обектите. Поведението се изразява в действията, които могат да бъдат извършвани от обектите. Средството на ООП, чрез което можем да описваме поведе­нието на обектите от даден клас, е декларирането на методи в класа.

Декларация на класа (class declaration) – това е редът, на който декларираме името на класа.

public class Dog

**3. Полета (fields)**

Променливи, декларирани в класа (някъде в лите­ратурата се срещат като член-променливи). В тях се пазят данни, които отразяват състоянието на обекта и са нужни за работата на методите на класа. Стойността, която се пази в полетата, отразява конкретното състояние на дадения обект, но съществуват и такива полета, наречени статични, които са общи за всички обекти.

// Field definition

private string name;

**Свойства (properties)**

Характеристиките на даден клас. Обикновено стойността на тези характеристики се пази в полета. Подобно на полетата, свойствата могат да бъдат притежа­вани само от конкретен обект или да са споделени между всички обекти от тип даден клас.

// Property definition

private string Name { get; set; }

**4. Конструктор (constructor)**

Псевдометод, който се из­пол­зва за създа­ване на нови обекти.

public Dog()

{

      // ... Some code ...

}

**5. Методи (methods)**

Методите представляват именувани блокове програмен код. Те извършват някакви действия и чрез тях реализират поведението на обектите от този клас. В методите се изпълняват алгоритмите и се обработват данните на обекта.

// Class declaration

public class Dog

{     // Opening brace of the class body

      // Field declaration

      private string name;

      // Constructor declaration

      public Dog()

      {

            this.name = "Balkan";

      }

      // Another constructor declaration

      public Dog(string name)

      {

            this.name = name;

      }

      // Property declaration

      public string Name

      {

            get { return name; }

            set { name = value; }

      }

      // Method declaration

      public void Bark()

      {

            Console.WriteLine("{0} said: Wow-wow!", name);

      }

}     // Closing brace of the class body

**6. Статични методи и конструктори**

В езика C# методите могат да бъдат „привързани“ както към даден клас като цяло (статични методи), така и към дадена инстанция (елемент) на дадения клас (динамични методи). Когато метода бъде дефиниран с ключовата дума static пред дефиницията си, то той е статичен. Този метод работи не върху конкретен обект от класа, а върху данни от целия клас.

Статичният конструктор има достъп само до статични членове. Не може да съдържа или да има достъп до членове на екземпляра.

**7.Статични полета и своиства**

В ООП има специална категория полета и методи, които се асоциират с тип данни (клас), а не с конкретна негова инстанция (обект). Наричаме ги статични членове (static members), защото са независими от конкретните обекти. Статичните членове принадлежат на самия клас, а не на някой конкретен обект от дадения клас. Нестатичните членове са състояния и поведения на отделните представители на категорията. Статичните членове се използват без да има създадена инстанция на класа, в който са дефинирани. Статичните членове могат да бъдат класове, полета, методи, конструктори и свойства.

**8.Източници**

<https://anastasova-bg.com/wp-content/uploads/2021/06/%D0%A3%D1%80%D0%BE%D0%BA-%E2%84%968-%D0%9E%D0%B1%D0%B5%D0%BA%D1%82%D0%B8-%D0%B8-%D0%BA%D0%BB%D0%B0%D1%81%D0%BE%D0%B2%D0%B5-%D0%BE%D1%82-%D0%BE%D0%B1%D0%B5%D0%BA%D1%82%D0%B8.pdf>

<https://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D0%B5%D0%BA%D1%82%D0%BD%D0%BE_%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%B0%D0%BD%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5>

<https://introprogramming.info/intro-csharp-book/read-online/glava14-definirane-na-klasove/>