

Кафедра компьютерной инженерии и моделирования

конспект (резюме) лекции №2

«Классы, структуры, конструкторы, модификаторы доступа»

по дисциплине «Объектно-ориентированное программирование»

Выполнил:

студент 2 курса группы

ПИ-б-о-231(2)

Покидько Максим Сергеевич

Проверил:

Заведующий кафедрой

компьютерной инженерии и

моделирования Милюков В. В.

Симферополь, 2024

Основные концепции C#

Классы

Классы — это шаблоны для создания объектов, которые инкапсулируют данные (поля) и операции над ними (методы).

Пример класса:

```
public class MyClass
{
    private int number;
    public string name;

    public MyClass(int num, string name)
    {
        this.number = num;
        this.name = name;
    }

    public int Number
    {
        get { return number; }
        set { number = value; }
    }

    public void DisplayInfo()
    {
        Console.WriteLine($"Number: {number}, Name: {name}");
    }
}
```

Модификаторы доступа:

- **public** — доступен везде.
- **private** — доступен только внутри класса.
- **internal** — доступен только внутри текущей сборки.
- **protected** — доступен в классе и его наследниках.

Перечисления (Enum)

Перечисления создают набор логически связанных констант для упрощения работы с кодом.

Пример:

```
enum Days
{
    Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
}
```

- По умолчанию значения начинаются с 0, но можно задать начальную нумерацию вручную.
-

Методы

Методы выполняют действия над данными класса.

- **Перегрузка методов:** Один метод может иметь несколько вариантов с одинаковым названием, но разными параметрами.
- **Статические методы** принадлежат классу, а не его экземплярам.

Пример перегрузки метода:

```
public void PrintInfo(string name) { ... }  
public void PrintInfo(string name, int age) { ... }
```

Возврат значений:

- **ref** — требует присвоения значения входному параметру.
 - **out** — требует присвоения значения выходному параметру.
 - **in** — гарантирует передачу параметра без создания копии.
-

Инкапсуляция, Наследование и Полиморфизм

- **Инкапсуляция:** Защищает данные, делая их доступными через публичные методы.
 - **Наследование:** Позволяет одному классу наследовать поля и методы другого для повторного использования кода.
 - **Полиморфизм:** Одинаковые методы могут работать с объектами разных классов через общий интерфейс.
-

Кортежи

Кортежи позволяют хранить набор значений разного типа в одной структуре.

Пример:

```
var person = (name: "John", age: 30);  
Console.WriteLine($"Name: {person.name}, Age: {person.age}");
```