ФГАОУ ВО «КФУ им. В.И. Вернадского»
Физико-технический институт (структурное подразделение)

Кафедра компьютерной инженерии и моделирования

конспект (резюме) лекции №5

«Классы, структуры, конструкторы, модификаторы доступа. Базовые понятия и принципы ООП в C#»

по дисциплине «Объектно-ориентированное программирование»

Выполнил:
студент 2 курса
группы ПИ-б-о-231
Покидько Максим Сергеевич

Проверил:
Заведующий
кафедрой компьютерной
инженерии и моделирования
Милюков В. В.

Симферополь, 2024

Основные понятия ООП в C#

1. Объекты и классы

- **Класс** – это описание объекта, а **объект** – экземпляр класса.
- **Конструктор** – метод, используемый для инициализации объектов (экземпляров класса).

2. Наследование

- Ключевое слово **base** используется для доступа к членам базового класса из производного класса.
    - Применяется для вызова метода базового класса, переопределенного в производном классе.
    - Используется для инициализации конструктора базового класса.

3. Автосвойства

- Упрощают код, автоматически создавая геттеры и сеттеры:

```
public string Name { get; set; }
```

4. Кортежи

- Позволяют возвращать несколько значений из метода:

```
var tuple = GetValues();
```

5. Структуры

- **Структура** – это значимый тип, в отличие от классов, которые являются ссылочными типами.
- Данные структуры хранятся в стеке, что обеспечивает быструю работу и экономию памяти.
- В отличие от классов, структуры не поддерживают наследование.
- Пример структуры:

```
struct Person
{
   public string name;
   public int age;
   public void Print()
   {
      Console.WriteLine($"Имя: {name} Возраст: {age}");
```

```
    }
}
```

6. Модификаторы доступа

- **private** – доступен только в пределах класса.
- **protected** – доступен в пределах класса и его наследников.
- **internal** – доступен только в пределах текущей сборки.
- **public** – доступен из любого места.
- **protected internal** – совмещает функционал `protected` и `internal`.
- **private protected** – доступен только в пределах сборки и производных классов.
- **file** – модификатор C# 11, доступен только в текущем файле кода.

7. Статические методы и поля

- **Статические методы** относятся к классу, а не к его экземплярам. Пример:

```
public static void MyMethod() { }
```

8. Конструкторы

- Конструкторы могут быть перегружены для предоставления разных вариантов инициализации объектов.
- **Конструктор по умолчанию** создается автоматически, если других конструкторов нет.

9. Полиморфизм

- В C# можно переопределять методы базового класса в производных с помощью ключевого слова `override`.

10. Инкапсуляция

- Ограничивает доступ к данным класса с помощью модификаторов доступа, таких как `private` и `protected`.