Турсунов Баходурхон

#16. .NET Object-Oriented Programming (OOP). Abstract Class. Virtual, Override, New, Abstract.

**1. Что такое абстрактный класс и чем он отличается от обычного класса?**

Абстрактный класс похож на обычный класс. Он также может иметь переменные, методы, конструкторы, свойства. Единственное, про при определении абстрактных классов используется ключевое слово **abstract**.

Главное отличие абстрактных классов от обычных состоит в том, что мы **НЕ можем** использовать конструктор абстрактного класса для создания экземпляра класса

**2. Как получить доступ к абстрактному классу и зачем нужно наследование?**

Для получения доступа к абстрактному классу нужно создать класс, который наследуется от этого абстрактного класса. Наследование позволяет использовать и расширять функциональность, предоставляемую базовым(абстрактным) классом.

Зачем нужно Наследование :

Наследование позволяет использовать существующий код в новых классах, избегая дублирования.

Наследование позволяет добавлять или изменять функциональность базового класса в производных классах.

Наследование является основой для абстракции и полиморфизма, что позволяет обрабатывать объекты производных классов как объекты базового класса.

**3. Приведите пример наследования от абстрактного класса и создания производного класса. Укажите, как переопределить абстрактный метод в производном классе.**

**using** System;  
  
// Абстрактный класс  
**public** **abstract** **class** **Animal**  
{  
 // Абстрактный метод, который должен быть реализован в производных классах  
 **public** **abstract** **void** **MakeSound**();  
}  
  
// Производный класс  
**public** **class** **Dog** : **Animal**  
{  
 // Переопределение абстрактного метода  
 **public** **override** **void** **MakeSound**()  
 {  
 Console.WriteLine("Woof! Woof!");  
 }  
  
 // Дополнительный метод, специфичный для класса Dog  
 **public** **void** **WagTail**()  
 {  
 Console.WriteLine("The dog wags its tail.");  
 }  
}  
  
**class** **Program**  
{  
 **static** **void** **Main**()  
 {  
 // Создаем объект производного класса  
 Dog myDog = **new** Dog();  
  
 // Вызываем метод, унаследованный от базового класса  
 myDog.MakeSound(); // Выведет "Woof! Woof!"  
  
 // Вызываем дополнительный метод класса Dog  
 myDog.WagTail(); // Выведет "The dog wags its tail."  
 }  
}

**4. Как ключевое слово "virtual" позволяет переопределять методы в производных классах?**

С помощью ключевого слова override и переопределенный метод в классе наследнике должен иметь тот же набор параметров, что и виртуальный метод в базовом классе.

**5. Можно ли использовать модификатор "virtual" с методами, которые являются статическими, абстрактными, приватными или переопределенными? Почему или почему нет?**

Нет, нельзя. Потому что компилятор ругается, если дописать какой-нибудь модификатор доступа рядом с статическим классом

Practice:

1. Представьте, что вы разрабатываете простую иерархию классов для зоопарка. Вам нужно представить различных животных и их характеристики. Для этого вы будете использовать абстрактные классы, виртуальные методы и наследование.

2. Создайте абстрактный класс "Животное" со следующими свойствами и методами:

3. Свойства: Имя (строка), Возраст (int)

4. Методы: MakeSound(): Этот метод должен быть помечен как абстрактный и не имеет реализации. Он представляет характерный звук животного. DisplayInfo(): Этот метод должен отображать имя, возраст и звук, издаваемый животным.

5. Создайте два производных класса от класса "Животное":

6. Класс "Лев": Реализуйте метод "MakeSound" для рева.

7. Класс "Слон": Реализуйте метод "MakeSound" для трубы.

8. Создайте экземпляры классов "Лев" и "Слон", задайте их имена и возраст.

9. Вызовите методы "MakeSound" и "DisplayInfo" для каждого экземпляра, чтобы наблюдать различные звуки, издаваемые животными, и отображать информацию о них.

10. Добавьте дополнительные свойства или методы в производные классы, чтобы представить специфические характеристики львов и слонов. Например, вы можете добавить свойство для представления цвета гривы льва или длины хобота слона.