1. **Что такое производный и базовый классы?**
2. **В чем заключена основная задача наследования?**
3. **Пусть базовый класс содержит метод basefunc(), а производный класс не имеет метода с таким именем. Может ли объект производного класса иметь доступ к методу basefunc()? Если да, то при каких условиях?**

Если у basefunc() модификатор доступа **не** private

1. **Напишите объявление конструктора без аргументов для производного класса B, который будет вызывать конструктор без аргументов базового класса A.**

Public B():base()

{

}

1. **Что такое полиморфизм? Приведите пример.**
2. **Определите назначение виртуальных функций.**
3. **Кому доступны переменные с модификатором protected?**

Переменные с таким модификатором доступны внутри класса, а также его наследником, в том числе и наследникам в других сборках.

1. **Наследуются ли переменные с модификатором private?**

Да

1. **As, is – что это, как применяется?**

С помощью этих операторов происходит проверка возможности приведения объекта к другому типу. Is возвращает значение bool. As, если возможно преобразование, возвращает преобразованный объект, если преобразование невозможно – возвращает null.

1. **Поддерживает ли C# множественное наследование?**

Множественное наследование классов не поддерживается, но имеется возможность реализовать в классе несколько интерфейсов.

1. **Можно ли запретить наследование от класса?**

Да, с помощью модификатора sealed.

1. **Можно ли разрешить наследование класса, но запретить перекрытие метода?**

Да, с помощью модификатора sealed

1. **Что такое абстрактный класс?**
2. **В каком случае вы обязаны объявить класс абстрактным?**
3. **В чем разница между абстрактными и виртуальными классами? Между виртуальными и абстрактными методами?**
4. **Какие компоненты класса могут быть виртуальными?**
5. **Что такое интерфейс?**

Интерфейсы позволяют определить некоторый функционал, не имеющий конкретной реализации. Затем этот функционал реализуют классы, применяющие данные интерфейсы.

1. **Как работать с объектом через унаследованный интерфейс?**
2. **Приведите пример явной реализации интерфейса.**

Interface IDno

{

Void WriteCode();

}

class Programmer : IDno

{

Public void WriteCode()

{

Console.Writeline(“Пишу код так скозать”)}}

1. **Почему нельзя указать модификатор видимости для методов интерфейса?**

Еще один момент в объявлении интерфейса: все его члены - методы и свойства не имеют модификаторов доступа, но фактически по умолчанию доступ **public**, так как цель интерфейса - определение функционала для реализации его классом. Поэтому весь функционал должен быть открыт для реализации.

1. **Можно ли наследовать от нескольких интерфейсов?**

Да

1. **Назовите отличия между интерфейсом и абстрактным классом.**
2. **Для чего используются стандартные интерфейсы ICloneable, IComparable, IComparer, lEnumerable?**
3. **В какой строке приведенного ниже фрагмента листинга не содержится ошибки?**

1, т.к. невозможно создать тело у абстрактного метода

3, т.к. модификатор new повторяется

4, т.к. статический метод не может быть помечен как virtual

1. **Что будет выведено на консоль в результате выполнения следующего фрагмента?**

3 4

1. **Что будет выведено на консоль в результате выполнения следующего фрагмента кода?**

A

B

1. **Чем может быть M4 если дано следующее определение?**

Только интерфейсом

1. **Выберите верное присваивание для объектов, определенных в листинге.**

Варианты ответа:

1) b = a; //не удаётся неявно преобразовать

2) a = b;

3) c = a;

4) d = a;

5) с = b;

1. **Что будет выведено на консоль в результате выполнения следующего фрагмента, если раскомментировать строчку 1?**

AB

1. **В какой строке приведенного ниже фрагмента листинга содержится ошибка?**

2, т.к. член помеченный как override не может быть помечен как new или virtual

1. **Почему приведенный ниже фрагмент листинга содержит ошибку?**

Нельзя создать экземпляр абстрактного класса.

1. **В какой строке может быть ошибка компиляции?**

Ошибки нет

1. **Что будет выведено на консоль в результате выполнения следующего фрагмента листинга?**
2. New G