1. **Назовите принципы ООП. Поясните каждый из них.**

Инкапсуляция – сокрытие кода и данных, позволяет исключить вмешательство извне.

Наследование – можно строить новые определения классов на основе предыдущих

Полиморфизм – он обозначает способность языка трактовать связанные объекты в сходной манере.

1. **Назовите класс .NET, от которого наследуются все классы.**

Object.

1. **Охарактеризуйте открытые методы System.Object.**

ToString(), GetHashCode(), GetType(), Equals()

1. **Охарактеризуйте закрытые методы System.Object.**

Finalize(), MemberwiseClone()

1. **Приведите пример определения класса.**

public class Student

{

public string name;

public string faculty;

public int group;

}

1. **Какие ключевые слова можно использовать при определении класса?**

class

1. **В чем отличие между объектом и классом?**

Класс — «коробка» для полей, свойств, объектов и методов.

Объект — экземпляр класса.

1. **Что такое конструктор? Когда вызывается конструктор?**

Конструктор — специальный метод, образец для создания объекта.

Конструкторы вызываются при создании нового объекта.

1. **Перечислите свойства конструктора?**
2. **Что такое деструктор (destructor) ?**

Деструктор — метод для освобождения неуправляемых ресурсов.

Задаётся при помощи знака ~.

1. **Что такое this?**

Ключевое слово, обеспечивающее доступ к текущему экземпляру класса.

1. **Что будет выведено в результате выполнения**

5 5

7 7

1. **Какие спецификаторы доступа для класса и методов класса существуют в C#?**

Public, private, protected, internal, protected internal, private protected

1. **Опишите модификатор protected internal.**

Классы и члены класса с таким модификатором доступны из текущей сборки и из производных классов.

1. **Зачем и как используются ref и out параметры функции?**

ref используется для передачи ссылки на переменную. Out работает как и return, позволяет получить значение из метода. Указываются и при вызове метода.

1. **Приведите пример необязательных и именованных параметров метода.**

Public static func\_sum(int a, int b = 10, int c = 20) {…}

Public static func\_max(int a, int b, int c){…}

Int max = func\_max (c:3, b:2, a:1);

}

1. **Приведите пример полей класса – статические, константные, только для чтения.**

static int primer\_static

const int primer\_const

readonly int primer\_readonly

1. **Приведите пример определения свойств класса. Как свойства связаны с инкапуляцией?**

class Student

{

    private string name;

    public string Name { get { return name; } set { name = value; } }

}

Свойства позволяют ограничить доступ к свойству.

1. **Назовите явное имя параметра, передаваемого в метод set свойства класса?**

value

1. **Что такое автоматические свойства?**

{get; set;} — сокращённая форма

1. **Что такое индексаторы класса? Какие ограничения существуют на индексатор?**

Индексаторы позволяют индексировать объекты и обращаться к ним по индексу.

1. **Что такое перегруженный метод?**

Несколько методов с одинаковым названием.

1. **Что такое partial класс и какие его преимущества?**

Позволяет записать определения класса в разные файлы, которые будут скомпилированы в один класс.

1. **Что такое анонимный тип в C#?**

Тип, который не принадлежит конкретному классу

var primer = new {…}

1. **Для чего делают статические классы?**

Статический класс может использоваться как обычный контейнер для наборов методов, работающих на входных параметрах, и не должен возвращать или устанавливать каких-либо внутренних полей экземпляра.

1. **В чем отличие статического поля от экземплярного?**

Доступ к статическому полю, в отличие от экземплярного, производится через имя класса, а не объект.

1. **Поясните работу статических конструкторов.**

Статические конструкторы не должны иметь модификатор доступа и не принимают параметров. В них нельзя использовать ключевое слово this. Статические конструкторы нельзя вызвать вручную, они вызываются сами при первом создании объекта.

1. **Какая разница между поверхностным (shallow) и глубоким (deep)**

**копированием?**

При глубоком копировании новый объект хранит копии значений объекта, тогда как при поверхностном копировании новый объект хранит ссылки на исходный адрес памяти.

1. **В чем разница между равенством и тождеством объектов?**

Тождество — равенство ссылок на объекты.

1. **Что такое частичные классы и частичные методы?**

Частичный класс — класс, который определён в нескольких файлах, но будет скомпилирован, как один. Определение частичного метода может находится в одном классе, а реализация — в другом. Задаются они ключевым словом partial.

1. **Что будет выведено на консоль результате выполнения следующего кода:**

System.Int32

1. **Что будет выведено на консоль результате выполнения следующего кода:**

a=2, b=1

1. **Пусть задан следующий класс.**

Какой из конструкторов задан неверно?

2

1. **Пусть задан следующий класс.**

Сколько аргументов может быть задано при вызове конструктора

данного класса?

От 0 до 2

1. **Почему не удается создать объект класса A?**

В телах конструкторов не записаны поля объекта.

1. **Что будет выведено в консоль при выполнении данной программы?**

A static

A

1. **Какая строка приведенного далее класса вызовет ошибку компиляции?**

3