1. **Что такое наследование?**

* Наследование (Inheritance) касается способности языка позволять строить новые определения классов на основе определений существующих классов.
* Наследование позволяет построить новый класс на основе имеющегося, добавив в него новые поля и методы.

1. **Какие классы называется базовыми и производными?**

* Базовый класс (предок) — класс, на основе которого строится определение нового класса - производного класса (потомка).

1. **Сколько предков может иметь класс?**

* Неограниченное количество

1. **Что такое множественное наследование? Чем оно опасно? Когда его можно использовать?**

* Множественное наследование - наследование от нескольких классов одновременно.
* При множественном наследовании возникает проблема неоднозначности из-за совпадающих имен в базовых классах.

Поэтому лучше наследоваться от интерфейсов и классов-контейнеров.

* Проблема ромба – когда два наследника и при вызове метода, он не знает какой из двух использовать.
* Устранение неоднозначности - в наследнике указывать к какому классу будет принадлежать свойство так A::prop = a, B::get()

Ромбовое наследование - решение, просто указать из какого класса вызываем: c1.B1::func();

1. **Что такое перегрузка метода? Что такое переопределение метода?**

* Перегрузка методов (overloading) – это объявление в классе методов с одинаковыми именами при этом с различными параметрами.
* Переопределение метода (overriding) - объявление в производном классе метода, который заменяет собой одноименный метод базового. При этом новый метод должен иметь те же параметры что и метод базового класса.

1. **Как работает преобразование типов, связанных наследованием?**

* Преобразование значения одного типа в значение другого типа.
* Выделяют приведения типов: явные, неявные.
* Явное приведение задаётся программистом в тексте программы с помощью:

1. конструкции языка
2. функции, принимающей значение одного типа и возвращающей значение другого типа.

* Неявное приведение выполняется транслятором (компилятором или интерпретатором) по правилам, описанным в стандарте языка.

1. **Что такое интерфейс (ООП)?**

* Интерфейс класса = способы взаимодействия с этим классом = методы
* Абстрактный класс без полей, с абстрактными (без реализации) методами.
* Это набор методов класса, доступных для использования другими классами
* Интерфейсы позволяют наладить множественное наследование объектов и в то же время решить проблему ромбовидного наследования.
* В языке C++ она решается через наследование классов с использованием ключевого слова virtual.
* Объявление класса нужно помещать в заголовочный файл. Клиент, желающий использовать этот класс, должен включать этот файл используя include. Это – интерфейс.

1. **Как влияют области видимости внутри класса на наследуемые методы и поля?**

* Наследование с ключом доступа Public:   
  public элементы базового класса становятся public элементами производного класса,   
  protected элементы базового класса становятся protected элементами производного класса.   
  Private элементы никогда не бывают доступны из производного класса, но могут быть доступны, с помощью вызова public и protected элементов базового класса.
* Наследование с ключом доступа Protected:   
  public и protected элементы базового класса становятся protected элементами производного класса.
* Наследование с ключом доступа Private:   
  public и protected элементы базового private класса становятся private элементами производного класса.
* Если при наследовании не указан спецификатор доступа, то типом по умолчанию private