1.Что такое классы и объекты?

Класс является логической конструкцией, на основе которой построен весь язык Java. Важная особенность класса состоит в том, что он определяет новый тип данных, которым можно воспользоваться для создания объектов этого типа, т.е. класс — это шаблон (чертеж), по которому создаются объекты (экземпляры класса

объект - экземпляр класса, и для того, чтобы с ним работать, его нужно положить в переменную, которой присваивается имя. На самом деле, в переменной не лежит весь объект, а только ссылка на него.

У объекта есть поля name, color, age (можно присвоить и другие) с которыми можно работать (получать или изменять их значения). Для доступа к полям объекта служит операция-точка, которая связывает имя объекта с именем поля.

2. Что такое инкапсуляция?

Инкапсуляция позволяет управлять доступом к членам класса из отдельных частей программы, предоставляя доступ только с помощью определенного ряда методов, что позволяет предотвратить злоупотребление этими данными и связывает данные с манипулирующим ими кодом.

3. Какие модификаторы доступа существуют и зачем они нужны?

В Java бывают следующие модификаторы доступа:

- public

- private

- protected

- также уровень доступа, предоставляемый по умолчанию.

Модификатор доступа предшествует остальной спецификации типа члена.

4. Как получить доступ к private полю объекта?

Компонент, объявленный как **private**, доступен только внутри класса, в котором объявлен. Если в объявлении члена класса отсутствует явно указанный модификатор доступа (**default**), то он доступен для подклассов и других классов из данного пакета. Если же требуется, чтобы элемент был доступен за пределами его текущего пакета, но только классам, непосредственно производным от данного класса, то такой элемент должен быть объявлен как **protected**.

5. Что такое наследование и зачем оно нужно?

Наследование позволяет создать класс (суперкласс), определяющий какие-то общие черты набора классов, а затем этот общий класс может наследоваться другими, более специализированными классами (подклассами), каждый из которых будет добавлять свои особые характеристики.

6. Что такое полиморфизм?

Полиформизм - это подход, при котором в переменной, имеющей тип родителя, допускается хранение ссылки на его потомка. Соответственно допускается использование потомка там, где по сигнатуре ожидается родитель, причём в случае "подмены" при попытке вызова какого - либо метода у родителя будет вызван метод потомка. Таким образом полиморфизм позволяет абстрагироваться от конкретных реализаций методов.

7. Что такое конструкторы, зачем они нужны, как взаимодействуют конструкторы родителей и наследников?

Конструкторы предназначены для инициализации объектов при создании в Java.

Имя конструктора обязательно должно совпадать с именем класса, а синтаксис аналогичен синтаксису метода. Если создать конструктор класса Cat как показано ниже, он автоматически будет вызываться при создании объекта.

8. Что такое абстрактные классы и методы?

Суперкласс, в котором нужно объявить структуру заданной абстракции, не предоставляя полную реализацию каждого метода. Для этой цели служит абстрактный метод (с модификатором **abstract**). Класс, содержащий хоть один абстрактный метод, должен быть объявлен как абстрактный (в объявлении класса также добавляется ключевое слово abstract).

9. Что такое статические поля и методы?

**Статические методы** — это методы в Java, которые можно вызывать без создания объекта класса.

Аналогично тому, как статический метод не привязан к объекту, можно объявить статическое поле, не являющееся частью никакого объекта, которое будет создаваться в одном экземпляре и будет доступно из любого объекта класса или через имя класса.

10. От какого класса наследуются все классы?

Класс, который наследует свойства другого класса, называется подклассом (производным классом, наследующим классом), а класс, свойства которого наследуются, известен как суперкласс (базовый класс, родительский класс)