Вариант 3

1) Что такое абстрактный класс?

2) В чем отличие интерфейса от абстрактного класса?

3) Каким образом в Java реализуется композиция?

4) Что такое статистический и динамический полиморфизм в Java?

5) Каково значение ключего слова super в Java?

1) Абстрактный класс - это класс, который содержит абстрактные методы, т.е. методы без реализации, а также может содержать обычные методы с конкретной реализацией.

Объекты абстрактного класса не могут быть созданы напрямую, но он может иметь подклассы, которые реализуют абстрактные методы. Абстрактный класс обычно используется для определения общего функционала, который может быть уточнен и расширен в подклассах.

Для того чтобы определить класс как абстрактный, используется ключевое слово abstract. Сам по себе абстрактный класс может содержать методы с реализацией, поля и конструкторы, но нельзя создать его экземпляр.

2) Интерфейсы используются для определения контрактов, которые класс должен реализовать, в то время как абстрактные классы используются для создания классов, которые содержат общую реализацию для подклассов

Отличия:

В интерфейсе можно только объявлять методы, но не предоставлять их реализацию, абстрактный класс может содержать как абстрактные методы (методы без реализации) так и конкретные методы (методы с реализацией). реализация

В Java класс может реализовывать несколько интерфейсов, но может наследоваться только от одного абстрактного класса.

Интерфейс не содержит переменные-поля, может содержать только константы, абстрактный класс может содержать переменные-поля, константы и другие члены данных.

Абстрактный класс может расширять другой класс и реализовывать интерфейс, интерфейс может только наследовать другой интерфейс.

3) В Java композиция реализуется путем создания объектов одного класса в другом классе. Объекты, которые содержат другие объекты, называются составными объектами, и этот подход позволяет создавать более сложные структуры, объединяя простые объекты в более сложные.

4) Статистический (или компиляционный) полиморфизм в Java проявляется через перегрузку методов и полиморфизм в методах наследования. Во время компиляции компилятор Java решает, какой метод вызывать, основываясь на типе ссылки на объект.

Динамический (или выполнение) полиморфизм в Java проявляется через переопределение методов в классах-наследниках. Во время выполнения программы Java выполняет переопределенные методы, основываясь на типе объекта.

Статический полиморфизм решается во время компиляции, а динамический полиморфизм проявляется во время выполнения программы.

5) В Java ключевое слово super используется для обращения к членам суперкласса (родительского класса) в подклассе (наследнике). Это позволяет подклассу обращаться к полям, методам, и конструкторам, определенным в его родительском классе.

Таким образом, super позволяет подклассу получать доступ к членам суперкласса и обеспечивает более гибкую работу с наследованием в Java.

Super используется для вызова конструктора суперкласса из конструктора подкласса, для вызова переопределенного метода из суперкласса.