1. Какво е Композиция - Използване на обекти от съществуващи класове в новия клас. Използва се многократно функционалността на съществуващия код.

2. Какво е наследяване - **Новият клас се създава като тип на съществуващ клас. Използва се формата на съществуващ клас, като добавяме към нея код, без да променяме съществуващия клас**

3. Какво е Полиморфизъм - **Свойството на обектите от един и същи тип да имат един и същи интерфейс, но с различна реализация на този интерфейс.**

4. Разлика между композиция и наследяване - **Композиция се използват обекти от съществуващи класове в новия клас.**

**Наследяване се използва при преобразуване нагоре.**

5. Какво е късно свързване – **Ако се изпрати съобщение до обект извикваният код не се определя до времето за изпълнение. Java използва специален код, който изчислява адреса на тялото на функцията, използвайки информация, съхранявана в самия обект**

6. Какво е свързване – **Компилаторът генерира обръщение към определено име на функция, програмата за свързване го преобразува в абсолютен адрес на кода, който трябва да се изпълни**

7. Късно свързване на базата на какво се прави – **На базата на информация**

8. Като създаваме един обект можем ли да променим референцията при наследяване - **Новосъздадения обект обикновенно не се присвоява на променлива от тип съвпадащ с класа на обекта. В променливата се запазва само референцията към новосъздадения обект**

9. Обяснете какво е поведението на конструктора - **Конструктора заделя памет в хийпа, където ще се съхраняват данните които се пазят в полетата на конкретния обект, инициализира всяко поле с подразбираща се за типа му стойност и връща референция към новосъздадения обект**

10. Взаимодействие между конструктори – **Съществува механизъм, чрез който един конструктор да извиква конструктор деклариран в същия клас. Извикването става с помощта на this(параметри)**

11. Какво е вътрешен клас – **Един клас може да се дефинира в друг клас. Обект от вътрешен клас може да бъде създаден само във връзка с обект от прилежащия външен клас**

12. Отношение между вътрешен и външен клас -  **В един самостоятелен външен клас може да има няколко вътрешни класове, всеки от които допълва същия интерфейс или наследява от същия клас по различен начин**

13. Има ли вътрешен клас достъп до данни на външен клас - **Вътрешните класове могат да съществуват и без външния клас, в който са разположени, но нямат достъп до неговата this инстанция.**

14. Какво е контролна рамка чрез вътрешния клас - **Структура за съхраняване на обекти от тип “Event”**

15. Принцип на идентификация на типовете ? - **имаме обект от тип class, когато се създава за първи път от класа , системата създава един обект, който е от клас class, той дава шаблона, който се използва.**

16. Каква е идеята на генерирането програмиране – **Искаме да правим универсални структури, които да са еднотипни (Stack)**

17. Какво е генетично програмиране**- В основата си терминът означава параметризирани типове. Параметризираните типове са важни те позволяват създаване на класове, интерфейси и методи , в които типът данни, с които те работят , е определен като параметър. Клас , интерфейс или метод , който работи с параметър за тип, се нарича генетичен: генетичен клас и генетичен метод.**

18. Какво е функционално програмиране- **Ламбда израз – нова езикова конструкция, води към нов (за Java) стил на програмиране, познат като “функционално програмиране“ . С нейна помощ, някои решения могат да бъдат формулирани по един елегантен начин .Особено предоставят предимства в областта на:**

**- Рамки – напр. колекции (Collections-Framework)**

**-Паралелна обработка.**

**Ключът към разбирането на ламбда са две конструкции :**

**-Първата е самата ламбда .**

**- Втората е функционалният интерфейс.**

**Един ламбда израз по същество е анонимен метод по този метод обаче не се изпълнява сам и вместо това се използва за изпълнение на метод, дефиниран от функционален интерфейс. Така, един ламбда израз резултира до една форма на анонимен клас. Ламбда изразите често се наричат затваряне (closure).**

**Ламбда: контейнер на първичен код, подобен на един метод, обаче без:**

**- Име**

**- Явно задаване на тип на резултата**

**- Възможни изключения**

**- Т.е. анонимен метод със следния синтаксис:**

**(списък параметри) → { израз или оператори }**

**ФУНКЦИОНАЛНИ ИНТЕРФЕЙСИ съществено ламбда израз може да се използва само в контекст, в който е специфициран целеви тип с ключовият аспект на функционалния интерфейс е, че може да се използва с всеки ламбда израз, който е съвместим с него.**

1. Може ли да дефинираме метод? - **Да**

+ Създаване и използване на обекти - **Създаването н става чрез new. Извикването и достъп до полетата става с оператора „ . ”**

+ Разликите между интерфейс и абстрактен клас - **Абстрактен клас - Това са методи, които съдържат само декларация, но нямат имплементация . Интерфейса може да съдържа абстрактни методи и член променливи**

+ Какви методи и променливи познавате в джава - **Методи на обекти (инстанции), Методи на класове (статични), Генетични методи, Не-генетични методи**

**Променливи - local class/static, instance**

+ Композиция – **Използване обекти от съществуващи класове в новия клас**

+ Възможно ли е в един fail .java да декларираме повече от един клас? – **Да**

+ Каква става променливата като я поставиш след final? – **Константа ако final стои пред клас не може да се наследява**

+ Какво означава Implements? – абстрактен метод. **Interface има един единствен runnable метод Implements.**

+ Възможно ли е множествено наследяване в Java да реализира множествено наследяване на класове>? – **Не**

+ Може ли в Java да се реализира механизъм за множествено наследяване? – **Да като се имплементират няколко интерфейса.**

+ Как се реализират нишки в Java? – **като наследим класа Thread и като наследим run \_\_\_\_**

+ Може ли да се изпълни sleep извън try cash box? – **Не, свърши ли try cash box-a нишката е свършила своята работа.**

+ Кои са модификаторите за достъп и за какво служат? – **default(по-подразбиране), public(общо достъпен), private(ограничен достъп), protected (защитен достъп)**

+ Типове методи? – **void (не връща стойност) – тип. Методите винаги започват с малки букви имат отваряща и затваряща скоба пример: $sectionName(Type, Name, ------), докато променливите са също с малка буква, но след тях няма скоба.**

+ Видове класове? **Integer, Boolean, Float, Double, Char(с единична кавичка ‘ ‘ ), String (с двойна кавичка “ ”)**

+ Каква е разликата между декларация и инициализация? – **Инициализация задаваме стойност, при декларацията създаваме променлива, но може да и зададем стоиността по-късно.**

**Какво е ключовата дума super ?**

Ключовата дума super се отнася до обекти от суперклас (родител).

Използва се за извикване на методи на суперклас и за достъп до конструктора на суперклас.