

Tentamensuppgifter

Exempeluppgifter – kan ändras

Uppgift 1

Skapa ett gränssnitt (interface), som innehåller ett antal metoder.

Skapa minst två klasser som implementerar detta gränssnitt.

Skapa en vektor med objekt, där det bara går att lagra objekt av de två klasserna.

Lagra flera objekt i den här vektorn.

Gå genom vektorn och anropa en metod från gränssnittet i samband med objekt i vektorn. Vilken kod som exekveras då?

Uppgift 2

Skapa ett gränssnitt (interface), som innehåller ett antal konstanter.

Skapa en klass som implementerar detta gränssnitt.

Uppgift 3

Vad är polymorfism?

Vad är dynamisk bindning?

Ange ett exempel.

Uppgift 4

Skapa en klass, som innehåller ett antal abstrakta metoder.

Skapa minst två icke abstrakta subklasser till denna klass.

Skapa en vektor med objekt, där det bara går att lagra objekt av de två klasserna.

Lagra flera objekt i den här vektorn.

Gå genom vektorn och anropa en metod från superklassen i samband med objekt i vektorn. Vilken kod som exekveras då?

Uppgift 5

Vad kan ett gränssnitt (interface) innehålla?

Vad kan en abstrakt klass innehålla?

Vilken är skillnad mellan ett gränssnitt och en abstrakt klass?

Uppgift 6

Kan en klass ha flera subklasser? Ange ett exempel från standardbiblioteket.

Kan en klass ha flera superklasser?

Kan ett gränssnitt ha flera subgränssnitt? Ange ett exempel från standardbiblioteket.

Kan ett gränssnitt ha flera supergränssnitt.

Uppgift 7

Skapa ett gränssnitt och minst två subgränssnitt till det.

Uppgift 8

Skapa en klass och ett gränssnitt.

Skapa en subklass till klassen, som implementerar givna gränssnittet och gränssnittet `java.lang.Comparable`.

Uppgift 9

Skapa en metod som har en referens av ett standardgränssnitt som parameter.

Anropa den här metoden.

Uppgift 10

Skapa en klass, som har en referens av gränssnittet `java.awt.Shape` som sin instansvariabel. Denna referens initieras i klassens konstruktor.

Skapa ett objekt av klassen.

Uppgift 11

Skapa en klass, som har en vektor med referenser av gränssnittet `java.awt.Shape`. Den här vektorn initieras i klassens konstruktor.

Skapa ett objekt av klassen.

Uppgift 12

Skapa en metod som har en vektor med referenser av en obestämd typ som sin parameter (typparameter). Anropa den här metoden.

Uppgift 13

Skapa en metod som har en parameter av typen `java.util.List<T>` som sin parameter.

Anropa den här metoden.

Uppgift 14

Skapa en metod som har en flexibel parameter, så att det går att anropa metoden med objekt av alla möjliga typer.

Uppgift 15

Ange ett exempel av en klasshierarki i samband med grafiska komponenter.

Vilka metoder finns i de olika klasserna? Vilka metoder som ärvs?

Uppgift 16

Skapa en klass som implementerar gränssnittet `javax.swing.event.ListSelectionListener`.

Använd ett objekt av klassen på ett vettigt sätt.

(Samma fråga kan komma med andra lyssnargränssnitt: `MouseListener`, `MouseMotionListener`, `KeyListener`, `ChangeListener`, `ActionListener`, ...)

Uppgift 17

Skapa en inre, anonym klass som implementerar gränssnittet `java.swing.event.ChangeListener`.

Använd ett objekt av klassen på ett vettigt sätt.

Uppgift 18

Skapa ett program med flera trådar.

Hur kan en tråd definieras i en inre, anonym klass?

Uppgift 19

Hur kan en tråd stoppas ifrån en annan tråd?

Kan metoden `java.lang.Thread.stop()` användas för detta ändamål?

Uppgift 20

Vad kan hända när flera trådar samtidigt använder ett och samma objekt på ett osynkroniserat sätt? Ange ett exempel.

Hur kan användningen av objektet synkroniseras?

Uppgift 21

Vad är en synkroniserad metod och vad är ett synkroniserat block? Ange exempel.

Uppgift 22

Hur kan synkroniseringen optimeras? Ange ett exempel.

Uppgift 23

När flera trådar samtidigt använder flera objekt, kan ett dödläge uppstå. Ange ett exempel.

Hur kan ett dödläge undvikas?

Uppgift 24

När ett objekt inte är i ett lämpligt tillstånd, kan det hända att en tråd inte kan utföra en operation i samband med objektet. Men det kan hända att tråden får hjälp utifrån, från en annan tråd. Hur?

Ange ett exempel.

Uppgift 25

Vad är synkroniseringslås, och varför används de?

Vad är ett skrivlås och ett läslås?

Uppgift 26

Definiera via ett gränssnitt en kanal som kan användas för kommunikation mellan olika trådar i ett program.
Skapa en implementationsklass.

Uppgift 27

Hur kan två trådar kommunicera med varandra? Ange ett exempel.

Uppgift 28

Vad är en asynkron kommunikation, och vad är en synkron kommunikation?
Ange ett exempel.

Uppgift 29

Ett serverprogram kan betjäna flera klienter samtidigt. För varje ny klient skapas en ny tråd som betjänar den. Hur går det till?

Uppgift 30

Ett serverprogram kan ha en pool av trådar. Dessa trådar betjänar de klienter som ansluter sig. Hur går det till?