class student

{

static int add(int a,int b)

{

return a\*b;

}

static int add(int a,int b,int c)

{

return a+b+c;

}

}

class test

{

public static void main

{

system.out.println(student.add(2,3));

system.out.println(student.add(2,3,4));

}

by changing datatype of argument

class student

{

static int add(int a,int b)

{

return a\*b;

}

}

static double add(double a,double b)

{

return a+b;

}

}

void main()

{

system .out.println(student.add(2,5));

system.out.println(student.add(1.2,5.6));

}

}

METHOD OVERRIDE

if the child class has the same method

as declared in the parent class it is

known as method overriding in java

OR

if a child class provides the specific

implementation of the method that has been declared

by one of its parent class then it is known asmethod

overriding

method overriding is used to provide the specific

implementation of a method which is alreadyy

provided by the base class or super class

rules for method overriding-----

1. the method must have the same name as in the parent class

2. there must be an relation (inheritance)

3. the method must have the same parameter

as in the parent class

example of method overriding

wap to illustrate the use of java method overriding

class vehicle

{

void disp()

{

system.out.println("this is base class");

}

}

class bike extends vehicle

{

void disp()

{

system.out.println("this is derived class");

}

}

class test

{

void main()

{

bike b=new bike();

b.disp();

}

}

output-----

this is derived class

can we override static method

no

a static method cannot be overridden becxause it can be

provided by run time polymorphism

can we override java main method

no because it is also static

explain reallife example of java method overriding

consider a scenerio where bank is a class

that provides functionality to get the rate of interset

however the rate of interest varies acc

to the bank for

example sbi,icici and axis bank to

provide 8%,7%and 9%

rate of interest

JAVA PACKAGE

a java package is a group of similar type of classes,

interfaces and sub packages

package in java can be categorised in two form built in

package and user defined package

BUILT IN PACKAGES

javalang, awt, javax, swing, io, util, sql etc

the package keyword is used to create a package in java

syntax

package packagename;

package vedika;

public class student

{

public static void main()

{

systeem.out.println("this is first package");

}

}

HOW TO ACCESS PACKAGE FROM ANOTHER PACKAGE

there are total three ways to access the package from

outside the package

1. by using package name

if you use package.\* then all the classes and interfaces

of this package will be accessible but not the sub packages

the import keyword is used to make the classes and interfaces

of another package accessible to current package

example-

//save by A.java

package pack;

public A

{

public static void disp()

{

system.out.println("hi this is pack");

}

}

//save by B.java

package mypack;

import pack.\*;

class B

{

public static void main()

{

A obj=new A();

obj.disp();

}

}

EXCEPTION HANDLING

EXCEPTION IN JAVA

in java an exception is an event that distrupts the normal flow

of program. it is an object which is thrown at runtime

EXCEPTION HANDLING

it is a mechanism to handle the run time errors such as

ClassNotFoundException

SQLException

etc

pascal case means first character of every word will be capital

the main advantage of exception handling is to maintain the

normal flow of the application

WHY WE USE MULTIPLE CATCH BLOCKS

WHAT IS THE DIFFERENCE BETWEEN CHECKED AND UNCHECKED EXCEPTIONS

TYPES OF EXCEPTIONS

there are mainly two types of exceptions

1.checked exception- the class that directly inherits the throwable class

except runtime exception and error.

for example- I/O EXCEPTION, SQLEXCEPTION , ClassNotFoundException

2.unchecked exception- the classes that inherits the runtime

exception are known as unchecked exceptions for example-

arithmetic exception, null point exception, Indexout exception

JAVA EXCEPTION KEYWORDS

TRY- the try keyword is used to specify a block where we should

place an exception code

it means we cannot use try block alone

the try block must be followed by either catch or finally

CATCH- the catch block is used to handle the exception

it must be preceeded by try block which means we cannot use catch

block alone

it can be followed by finally block later

FINALLY-the finally block is used to execute the necessary code

of the program.

it is executed whether an exception occur or not

JAVA EXCEPTION PROGRAM

syntax of try and catch

try

{

// code that may throw an exception

}

catch(Exception\_Class\_Nmae ref)

{

--------

}

syntax of try and finally

try

{

--------------

}

finally

{

----

}

example--

public class trycatchexample

{

public static void main()

{

int x=50/0;

system.out.println("testing");

}

}

it will throw an exception

public class trycatchexample

{

public static void main()

{

try

{

int x=50/0;

}

catch(ArithmeticException a)

{

system.out.println(a);

}

system.out.println("testing");

}

}

JAVA FINALLY BLOCK

it is always executed no matter whether there is an exception o

or not

the finally block is optional and for each try block there can be

only one finally block

syntax of finally

try

{

code---

}

catch(expression ex)

{

catch block code...

}

finally{

finally code here

}

java exception handling using finally block

class main

{

void main()

{

try{

int x=5/0;

}

catch(ArithmeticException e)

{

system.out.println("ARITHMETIC EXCEPTION");

}

finally

{

system.out.println("this is finally block");

}

}

THROW AND THROWS

NOTE : java throw keyword is used to explicitly(manually) throw a

single exception

when we throw an exception the flow of the program moves from the

try block to the catch block

the throws keyword is used to declare the exceptions

it specifies that there may occur an exception in the method

it does not throw an exception

EXAMPLE OF THROW

class main

{

public static void DivideByZero()

{

throw new ArithmeticException("trying to divide by 0");

}

public static void main(string[] args)

{

dividebyZero();

}

}

Q-WHAT IS EXCEPTION HANDLING?

EXPLAIN THE DIFF WAYS TO HANDLE THE EXCEPTION

Q-EXPLAIN THE KEYWORDS USED IN EXCEPTION HANDLING

MULTI-THREADING

it is a process of executing multiple threads

simultaneously

a thread is a light weight sub process , the smallest unit

of processing

we can multi threading than multi processing because threads use

a shared memory area they do not allocate seperate memory area

ADVANTAGES OF MULTI THREADING

1. it does not block the user because threads are independent

and you can perform multiple operations at the same time

2. you can perform many operations together , so it saves time

3. threads are independent, so it does not affect other threads

if an exception occurs in a single thread

JAVA THREAD CLASS

java provides thread class to achieve thread programming

thread class provides constructors and methods to create

and perform operations on thread

DIFFERENT TYPES OF METHODS USED IN JAVA

1. void start()- it is used to start the execution of the thread

2. void run()- it is used to do an action for a thread

3. void sleep()- it sleeps a thread for a specified amount of time

4. void current()- it return a reference to the currently executing thread

5. void suspend()- it is used to suspend a thread , suspend means depreciated

6. void destroy()- it is used to destroy thread and all of its sub groups

7. void resume()- it is used to resume the suspended thread

8. void isalive()-

9. void getpriority()-

10. void setpriority()-

11. void activecount()-

12. void checkaccess()-

13. void holdlock()-

14. void notify()-

15. void notifyall()-

imp

16. void getdefaultuncaughtexceptionhandler()-

17. void setdefaultuncaughtexceptionhandler()-

18. void setcontextclassloader()-

19. void getcontextclassloader()-

in java a thread always exist in any one of the following states:

1. NEW -whenever a new thread is created, it is always in the new state

for a thread in a new state, the code has not been run yet

2. ACTIVE- when a thread invokes the start method, it moves from the new

state to the active state

3. BLOCKED/WAITING- whenever a thread is inactive for a span of time

(not permanently) then either the thread is in blocked state or it is in

waiting state

4. TIME WAITING- sometimes waiting for leads to starvation

example- a thread (name A has entered in the section of a code and not

willing to leave that section in such a scenario another thread(name B)

has to wait, which leads to starvation

5. TERMINATED- a thread reaches the termination state because of the

following reasons

1. when a thread has finished its job then it exist or terminate normally

2. abnormal termination- it occurs when some unusual event such as an unhandled

exception or default

note: a terminated thread means the thread is no more in the system in other

words the thread is dead

what is diff between runable and running active state

explain the block/waiting state of java thread by using the example

how to perform two task by two threads

how to perform multi threading by anonymous class

what happens if we start a thread twice

what happen if we call the run method instead of start method

what is the purpose of finalise method

what is deadlock and when it can occur

MULTI THREADING PROGRAM IN JAVA

we are going to create a thread and explore the built in methods

package demotest;

public class thread\_example

implements Runnable

{

public void main()

{

Thread guruthread1= new Thread();

guruthread1.start();

try

{

guruthread1.sleep(1000);

}

catch(InterruptedException e)

{

e.Printstacktrace();

}

WAP to display the concept of multi threading

Write a menu driven code for exception handling if user press

1 then show

the working of try and catch block only

if user press 2 then show the working of try catch and finally

WAP to show the working of method overriding

WAP to show the concept method overloading

JVM( java virtual machine)

jvm is an java virtual machine ,it provides run time

environment in which java byte code can be executed

note:

class loader is a sub system of jvm which is used to load

the class files

whenever we run the java program we need to load the class

loader first

THIS KEYWORD [imp]

HOW MANY BUILT IN INTERFACES ARE THERE IN JAVA

WHAT IS THREAD CLASS

HOW CAN WE INITIALIZE THREAD CLASS IN JAVA

WHAT ARE THE BUILT IN FUNCTIONS THAT WE USE IN

MULTITHREADING

DEFINE THE FOLLOWING TERMS:

1. BUILT IN DATA TYPES AND USER DEFINED DATATYPES

2. CONSTANTS AND LITERALS

3. KEYWORDS

WHAT ARE OPERATORS EXPLAIN UNARY, ARITHMETIC, SHIFT

,LOGICAL , RELATIONAL , BITWISE, TERNARY AND ASSINGMENT OPERATOR

EXPLAIN ANY THREE FEATURES OF OOPS IN JAVA

EXPLAIN JAVA COMMENTS AND ALL THE THREE TYPES OF COMMENTS

IN JAVA

APPLET

ADVANTAGES

there are many advantages of applet

1. it works at client side so less response time

2.it is much secured

3. it can be executed by browsers running under many platforms including linux

,windows and mac os

DISADVANTAGES

plug in is required at client browser to execute applet

explain the lifecycle of java applet[imp]

applet is initialized

applet is started

applet is in running

applet is stopped

applet is destroyed

public void init()

it is used to initialize the applet it is only invoked once

public void start()

it is invoked after the init method or browser is maximised

it is used to start the applet

public void paint()

it is used to paint the applet

it provides graphics , class object that can be used for drawing rectangle,square, arch etc

public void stop()

it is used to stop the applet

it is invoked when applet is stop or browser is minimized

public void destroy()

it is used to destroy the applet. it is invoked only once

packages required to run the applet

1. import java.applet.applet

2. import java.awt.graphics

program to create basic java applet

graphic class in java

this class is displaying graphics in applet

java.awt.graphics class provides many methods for graphic programming

there are few common methods of graphic class

1. public void drawstring()- void drawstring( string, int x, int y)

it is used to draw the specified string

public class main extends applet

{

public void paint(graphics g)

{

g,drawstring("welcome in java applet",40,20);

}

}

public void drawrect( int x, int y, int width, int height)

it draws a rectangle with the specific width or height

g.drawrect(20,5, 50,60);

what is the diff between web application and window application

windows application- it is installed in the windows platform using windows operating system

it can only run on windows platform

it is accessible only from system in which it is installed

web application- it is installed on web server

AWT( abstract window toolkit)

it is an API to develop graphical use or windows based application in java

JAVA AWT COMPONENT ARE platform dependent that is components are displayed acc to the

view of operating system

the java.awt package provides classes for awt API such as text field ,label, text area ,

radio button ,checkbox , choice ,list etc

what are the components of awt

1. all the element like button , text fields ,scroll bars are called components

2. in java awt in order to place every component in a particular position on a screen we need to

add them to container[window ,frame ,dialouge]

USEFUL METHODS

public void add(classname obj)- this is used to insert a component

public void setsize(int width, int height)- it sets the width and height of the

component

public void setlayout(layoutmanager m)- it defines the layout manager for the

component

public void setvisible(boolean status)- it changes the visibility and

by default it is set to false

note: to create simple awt example we need a frame and we have two ways

to create a gui[graphic user interface] using frame in awt

1.by extending frame class(by inheritance)

2. association

PROGRAM

import java.awt.\*;

public class AWTExample extends frame

{

AWTExample1()

{

button b=new button("click me");

b.setBounds(30,100,80,30);

add(b);

setsize(300,300);

setTitle("this is our basic awt example");

setLayout(null);

setVisisble(true);

}

public static void main

{

AWTExample f=new AWTExample();

}

}

a flowlayout arrange the components in a directional flow and it is default

container of applet

windowadapter class-

it is an abstract adapter class which is used to recieve the window events

what is the diff between add window listner and new window adapter

AWT PROGRAM FOR COUNTRY DROPDOWN

FILE HANDLING IN JAVA

in java a file is an abstract datatype A named location used to store related information

known as file . there are several file operations like:

creating a new file

getting information about file /reading data of file

writing into a file

BYTE STREAM

mainly involves byte data

a file handling process with a byte stream is a process in which an input is

provided and executed by data

CHARACTER STREAM

mainly involves character data

a file handling process with a byte stream is a process in which an input is

provided and executed by characters

SOME IMP METHODS IN FILE HANDLING

canread()-return type is boolean. it is used to check if we can read the data of a file

createnewfile()-return type is boolean. it creates an empty file

canwrite()- return type is boolean. it is used to check if we can write the data in a file

exist()-return type is boolean.it is used to check if a specified file exists or not

delete()-used to delete a file

getname()- returntype is string. used to find the filename

getabsolutepath()-returntype is string

absolute path refers to the complete detail of a file or folder

wap to write some data in a file

EVENTS AND LISTNERS

changing the state of an object is known as event for example

click on a button

dragging of mouse

key press key up and key down

the java.awt.event package provides many event classes and listner

interfaces for event handling

event classes listner interfaces

mouse event(built in class) mouselistner and mousemotionlistner

mousewheelevent mousewheellistner

keyevent keylistner

windowevent windowlistner

componentevent componentlistner

STEPS TO PERFORM EVENT HANDLING

1. register the component with the listners (for button, public void addActionListener(ActionListener a){})

JAVA EVENT HANDLING WITHIN THE CLASS

JAVA EVENT HANDLING OUTSIDE THE CLASS

import java.awt.\*;

import java.awt.event.\*;

class AEvent2 extends Frame{

TextField tf;

AEvent2(){

//create components

tf=new TextField();

tf.setBounds(60,50,170,20);

Button b=new Button("click me");

b.setBounds(100,120,80,30);

//register listener

Outer o=new Outer(this);

b.addActionListener(o);//passing outer class instance

//add components and set size, layout and visibility

add(b);add(tf);

setSize(300,300);

setLayout(null);

setVisible(true);

}

public static void main(String args[]){

new AEvent2();

}

}

import java.awt.event.\*;

class Outer implements ActionListener{

AEvent2 obj;

Outer(AEvent2 obj){

this.obj=obj;

}

public void actionPerformed(ActionEvent e){

obj.tf.setText("welcome");

}

}