JAVA CORE

KEYWORDS

OBJECT

is a real world entity, object also known as instance.
it has 3 characteristic,

identity: unique ID provided by the JVM used internally
state: the stored variables
behavior: the methods show the behavior of an object
example: dog 22 (id) with red hair(variable) is running (method)

FINAL

this is an access modifier. variable becomes fixed and cannot be altered
methods cannot be overriden
classes cannot be inherited

FINALLY

used in try catch block . It is the block present in a program where all the codes written inside it get executed irrespective of handling of exceptions

FINALIZE

Prior to the garbage collection of an object, the finalize method is called so that the clean-up activity is implemented

METHOD OVERLOADING

allows methods to have the same name but different input parameters

```
class OverloadingHelp {
   public int findarea (int l, int b) {
           int var1;
           var1 = l * b;
           return var1;
   }
   public int findarea (int l, int b, int h) {
           int var2;
           var2 = l * b * h;
           return var2;
   }
}
```

DEPENDENCY INJECTION

used with interfaces and annotation to provide more flexible code

```
public interface Client {
    void doSomething();
}

public interface Service {
    String getInfo();
}
```

```
public class ServiceB implements Servic

    @Override
    public String getInfo() {
        return "ServiceB's Info";
    }
}
```

STRING INMUTABLE

for security purposes and shared reference can be shared anywhere

COLLECTIONS

LIST INTERFACE
ArrayList, LinkedList
SET INTERFACE
Linked, HashSet
QUEUE INTERFACE

THIS

The main purpose of using this keyword is to solve the confusion when we have same variable name for instance and local variables.

```
class Demo
{
 Double width, height, depth;
 Demo (double w, double h, double d)
 {
  this.width = w;
  this.height = h;
  this.depth = d;
 }
 public static void main(String[] args) {
     Demo d = new Demo(10,20,30);
     System.out.println("width = "+d.width);
     System.out.println("height = "+d.height);
     System.out.println("depth = "+d.depth);
 }
}
```

TIBCO EMS

KEYWORDS

MESSAGING MODELS	Queue (Point-to-Point) where it goes to only one possible subscriber Topic (Publish and Subscribe) goes to each and every subsriber
STATIC QUEUES	created through tibco admin tool
DYNAMIC QUEUES	created on EMS server or designer
TEMPORARY QUEUES	exist until client and connection exists
BRIDGES	You can send the same the messages to different destinations (queue or topic) within the same server
ROUTES	You can send the same messagges to different destinations (queues or topics) to different servers

DESTINATIONS:

STATIC DESTINATION	Can be either queues or topics both either queue or topic and be either static or dynamic
DYNAMIC DESTINATION	Stored in a file until deleted and created through admin tool Short termed, not stored and do no appear in configuration file

DELIVERY MODES:

PERSISTENT DELIVERY	Stores message on the disk or db
NON-PERSISTENT	Message is not stored
RELIABLE DELIVERY	
DESTINATION NAMES	each element created by a dot separator (citiacct.bill_payment.)
DESTINATION PROPERTIES	exclusive: only available to queues, when set, is sent to one consumer, no other consumer can receive except for standby consumer expiration: sets expiration time overrided by 0 which means the message is not going to expire
CONNECTION FACTORY	holds parameters for EMS server (user, password, provider URL)
TRA	
FLOW CONTROL ON DESTINATIONS	Some times the producer may send messages faster than the consumers can receive them. So, the message capacity on the server will be exhausted. So we use flow control. Flow control can be specified on destinations.
JMS QUEUE REQUESTOR	The JMS Queue Requestor activity is used to send a request to a JMS queue name and receive a response back from the JMS client.

COMMANDS

tcp://EMS01:7022>create queue SAMPLE.QUEUE secure	creates queue
tcp://EMS01:7022>create topic SAMPLE.TOPIC secure	creates topic
tcp://EMS01:7022>I	shows server information
tcp://EMS01:7022>create user "user1" password=password	
tcp://EMS01:7022>create bridge source=topic:TECH.TOPIC target=queue:TECH.QUEUE Selector="JMSCorrelationID='SAM'	

FILES

tibemsd.conf	It is the main configuration file that controls the characteristics of the EMS server. Here you can set up fault tolerant with the flag ft_active
tibcoadmin.exe	tool to administer ems servers
Queues.conf Topics.conf Routes.conf Factories.conf stores.conf groups.conf, users.conf, transports.conf	

TIBCO BW

KEYWORDS

AliasLibrary resource allows you to specify aliases to file system resources (such as a .jar file) that need to be included in your project

Domain a network based group of computers. These computers, in a domain, share a common database on that network

AESchema	Active Enterprise Schema
ABSTRACT WSDL	Contains porttype (operations, input and output)
GROUPING ACTIVITIES- TYPE OF GROUP ACTIONS	Iterate, repeat until true, repeat on error until true, critical section, transaction, pick first, while true.
TRANSACTION GROUP	A set of activities which should behave as one
TRA	Tibco Runtime Agent, Supplies agent running in the background of each machine, the run-time environment that is all the shared libraries and third-party libraries.

ACTIVITIES AND PROCESS STARTERS

jdbc activity	used for DML (insert, update, modify)
SQL Direct activity	Used for dynamic DB operations
File poller process starter activity	Polls files or directories with the specified name and starts a process when creation, modification, deletion is detected
File adapter	is got more features then file poller where you can sychonize many files and set up transport when the changes of a file happen
Parse XML	will read xml content according to the xml schema
Render XML	will write an xml output according to the xml xchema

WSDL

ABSTRACT WSDL	contains <type> <message>, <porttype>
CONCRETE WSDL	the same elements as abstract but with <binding>, <service>
ROOT ELEMENT	<definintion>
SERVICE ELEMENT	Contains the URL of the webservice, protocol and port
BINDING ELEMENT	container for the different types of operations
PORTTYPE	container for the different types of operations

COMMANDS

FILES

.ear	contains .par, .sar and .aar (adapter archive) files
vcrepo.dat	This file located in the root folder is used to store properties such as display name, TIBCO Rendezvous encoding, and description.
bwengine.tra	contains claspaths of pallettes
.par	Process Archive

PALLETTES

FILE PALLETTE copy file, read file, write file

XML TOOLS

DOCKER

HTTPS://WWW.YOUTUBE.COM/WATCH?V=5EB3SN51WIG

KEYWORDS

hypervisor	software virtualization which allocates resources of th host and the guest system
Dockerfile	text file with commands used to build and run image
Docker compose	is a YAML consisting all the details regarding various services, networks and volumes needed for setting up the docker application
Docker namespace	is like a linux partition
Docker Registry	is where you store the images either public or private
Docker client	
Docker host	
Docker Registry	
Docker Hub	It is a public cloud based for storing public images and sharing
Volumes	

COMMANDS

https://youtu.be/j61yfEfeJAE?t=1865

docker pull [image url]	downloads an image
docker stop [container name]	stops container
docker run and **docker start**	docker run is to set up the image first time, docker start is used to start such image
docker network ls	list networks
docker ps -a	status of containers
docker logs [container name] -f	show logs in real time of the container
docker rm [container name]	removes container when being in use
docker network create	create network to connect to host
docker container rename [container] [new name]	rename container
docker inspect [network name]	show details of network in docker

you can open terminal in docker as well: click container name -> terminal
you can execute commands to the container like below once the container is running

launch this command to generate (reset) the elastic user password
docker exec -ti es-node01 /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic
execute command inside container -**i**nteractive with pseudo --**t**ty to generate enrollment token
docker exec -it [container name] /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana

docker run --name kibana02 --net elastic -p 50601:5601 docker.elastic.co/kibana/kibana:8.7.1

wsl -d docker-desktop	switch to the windows subsystem for linux in docker
sysctl -w vm.max_map_count=262144	set the memory heap for virtual machine

DOCKER-COMPOSE

docker-compose up -d	create and start cluster through .env and yaml file
docker-compose down -v	delete the network, containers, and volumes
docker-compose.yml	The instructions of which ports, images, and shell scripts that you can run automatically through this file instead of manually

SHELL SCRIPTING

#!	operator shebang which directs the script to the interpreter's location
pwd	Print Working Directory
find	find -type f\(-name "*.txt"\) finds files with different type extension

OPERATORS

-b checks if file is a block
-c checks if file is a character special file
-d checks if is a directory
-e checks if file exists
-r checks if file has read access
-w check if the file has write access
-x check if the file has execute access or not
-s checks the size of the given file

PRINT CONTROLS

\" double quote
\\ backlash
\b backspace
\e Escape
\n New Line
\r Carriage Return
\t Horizontal tab

KUBERNETES

KEYWORDS

POD smmalles unit of kb8s. Usually one application per pod. Each pod gets its own IP
SERVICE permanent IP address. Is a load balancer as well to route traffic
INGRESS The request goes first to ingress and then to the **service**
ConfigMap No need for rebuild the application and containner since is an external configuration file
SECRET Used to store secret data, base64 encoded
VOLUMES Data storage. It could be local, cloud or remote so it can be persistant
DEPLOYMENT Another abstraction on top of pods
StatefulSet Is used for databases

NODE PROCESSES
Kubelet Interacts with container and node. Starts the pod with a container inside
Kube Proxy Forwards the requests
CONTAIRNER RUNTIME

MASTER NODE
API SERVER: acts a gatekeeper

COMMANDS

minikube start --vm-driver=hyperkit

kubectl get nodes **get status of nodes**

minikube status **get status of nodes**

kubectl get pod **show pods**

kubectl get services

kubectl create deployment nginx-depl -image=nginx **show pods**

kubectl get deployment

SPLUNK

POD smmalles unit of kb8s. Usually one application per pod. Each pod gets its own IP
SERVICE permanent IP address. Is a load balancer as well to route traffic
INGRESS The request goes first to ingress and then to the **service**
ConfigMap No need for rebuild the application and containner since is an external configuration file
SECRET Used to store secret data, base64 encoded
VOLUMES Data storage. It could be local, cloud or remote so it can be persistant
DEPLOYMENT Another abstraction on top of pods
StatefulSet Is used for databases

NODE PROCESSES
Kubelet Interacts with container and node. Starts the pod with a container inside
Kube Proxy Forwards the requests
CONTAIRNER RUNTIME

MASTER NODE
API SERVER: acts a gatekeeper

DEVOPS

https://www.youtube.com/watch?v=WxjJlYFIWtI

SELF

tell me about yourself and what technologies you have used in the previous project?
My day to day tasks were to accept incidents in ServiceNow regarding different issues in our infrastructure. Sometimes we would have to analyze code and reproduce the issues in our environment, document it and send it to the developer. The developer would send back the code to us and we would push it to all lower environments with the help of git, jenkins, ServiceNow and RLM.
I would create rules/alerts in tibco hawk depending on the requirements of the client/tech lead. Such alerts would include low disk space, large memory/cpu usage and notify groups through emails if an instance was down.

what was your team size?

which is your role in the team?
I was a tibco support engineer since I would go from writing scripts for the alerts, deploy on different environments bug fixes and help users find their errors in their applications

how good are you with programming?
although I haven't fully developed a fully fledged enterprise application for a company, I consider myself good at programming since I can read code and debug code

LINUX

chrontab -l
command to view the chrontab

what is an alias in linux?
something that tells you the shortcuts on that system this are defined in etc/rc file

chmod
Ichanges permission of a file in linux (rwx)

what is ssh port forwarding?
it allows to bypass firewalls or stricted guarded environments and you can connect to your servers in your LAN

what are zombie processes?
these are ghost processes which is in a terminated state but has not released the resources. It's entry is on the process table

top command

DEPLOYMENTS

what is a blue-greey deployment?
take half of the running servers, update them, put them live and take the other half to update them and put them live. You never let the user see the downtime

hot deployments
you have two environments of the same size and then you redirect traffic through a load balancer to one of the environents. Deploy it to the other and redirect traffic to the other

what is your **rollback strategy?**
every deployment should have one. You have a jenkins or a job to see if the deployment was successful and check if the endpoints are also running

have you used jenkins for deployment?
Yes I have used it. We used it to compile our tibco code with tibco commands. Like compile, build, package and push it to the development domains Plugins like maven or gradle weren't accisible to our team but some other teams did use those plugins

what jenkins plugin have used?
Like I mentioned we only used the tibco plugin and git plugin to get the latest code from the repositories.

AWS QUESTIONS

what is the difference between public and private subnet?
a public subnet is directly accesible from the internet
a private subnet is only accessible from within the vpc

what is cloud formation?
it's an orchestration tool from AWS, or a server deployment tool

PRODUCTION SUPPORT

what is the biggest issue you have faced in production?

what is your DR strategy in a live website?
it's an orchestration tool from AWS, or a server deployment tool