JAVA

INTERVIEW

SORULARI

By Meryem OZGUN

ozmrym7@gmail.com

JAVA NEDİR ÖZELLİKLERİ VE AVANTAJLARİ NELERDİR?

Java bir programlama dilidir. Bizim codlarimizi compile(derleme) ederek binary system(1 ve 0 lardan olusan sisitem) e cevirir

Javanın avantajları:

* ogrenmesi kolaydır
* dünyada en çok kullanılan programlama dillerinden biridir
* güvenlidir
* oop dir (object oriented programing, obje tabanlı programlama)
* java independent tir(platformlardan bağımsız calisir)
* otomatik hafiza yönetimine sahiptir. bellek yönetimi basarilidir.

JDK JRE JVM NEDİR?

"JDK," "JRE," ve "JVM" kavramları, Java programlama dilinde sıklıkla kullanılan terimlerdir ve Java uygulamalarının çalıştırılmasında önemli rolleri vardır. İşte bu kavramların açıklamaları:

1. **JDK (Java Development Kit):** JDK, Java geliştirme kitinin kısaltmasıdır. JDK, Java programlarını yazmak, derlemek ve çalıştırmak için gereken tüm araçları ve kütüphaneleri içerir. Bir Java geliştiricisi, JDK'yı kullanarak Java programları oluşturabilir, Java kodunu derleyebilir (Java byte koduna dönüştürebilir) ve ardından bu byte kodunu çalıştırabilir. JDK aynı zamanda Java SE (Standard Edition) ve diğer Java platformları için gereken tüm araçları ve kütüphaneleri içerir.
2. **JRE (Java Runtime Environment):** JRE, Java çalışma ortamının kısaltmasıdır. JRE, Java uygulamalarını çalıştırmak için gerekli olan temel bileşenleri içerir. Bir kullanıcı, bir Java uygulamasını çalıştırmak istediğinde, bilgisayarında JRE'nin kurulu olması gerekir. JRE, Java Virtual Machine (JVM) ve Java sınıf kütüphanelerini içerir ve bu sayede Java uygulamalarını çalıştırabilir.
3. **JVM (Java Virtual Machine):** JVM, Java sanal makinesinin kısaltmasıdır. JVM, Java uygulamalarını çalıştırmak için sanal bir ortam sağlar. Java programları, JVM üzerinde çalıştırılır ve JVM, platformdan bağımsız bir şekilde Java byte kodunu gerçek makine koduna dönüştürür ve çalıştırır. Bu, Java'nın platform bağımsızlığını sağlayan temel özelliktir. Farklı işletim sistemlerinde çalışabilen JVM, Java uygulamalarının aynı şekilde çalışmasını ve taşınabilir olmasını sağlar.

PROCEDURAL PROGLAMA VE NESNE YÖNELİMLİ PROGRAMLA ARASİNDAKİ FARK NEDİR?

Procedural programlama -> sistem tabanlı programlamadır. Sistem güzel olursa hersey güzel olur.örnegin bir okulda ogretmen ve müdür cok başarılı ise ordan başarılı ogrenciler cıkar.

Object oriented programla -> obje tabanlı programlama.obje mükemmel ise sistem tikir tikir isler. Kusursuz objeler olusursa görevler eksiksiz yapililr ve app ler kusursuz calisir. Ornegin: son sistem teknoloji ile donatılmıs harika bir mimariyle yapılmıs ,siniflarinda her türlü arac gerecin olduğu okuldan basarılı ogrenciler cıkar.

Gerçek dünyada prosedural programlama geçerlidir ve başarılıdır . sanal dünyada ise oop daha başarılıdır.

JAVADA PACKAGE NEDİR?

Bir projede birbiriyle ilgili classların bir arada tutulduğu yer .avantajları;

İsim catismasini önler

Kodlara kolay ulasim sağlar

Kategarizasyon sağlar

Gizli classlari barindırır baksa package den ulasımın engellendiği clasları barindirir.default acces modifier ile

CLASS NEDİR?

Javada class kaliptir. Üretmek istediğimiz objeleri bu kaliplara göre üretiriz..diğer bir deyisle javaya ne oluşturacağımızı bildirmek olusturmak istediğimizin seyin tipi

İçinde field (veriable) ve methodların bulunduğu calisma alanımız.

OBJE NEDİR?   
Java'da "obje" (object), nesne tabanlı programlamanın temel yapı taşıdır. Bir obje, belirli bir sınıfa ait bir örnektir ve bu sınıfın özelliklerini (alanlarını) ve davranışlarını (metodlarını) içerir. Java'nın nesne tabanlı programlama (OOP) yapısında, her şey bir obje olarak düşünülür ve uygulamalar objeler arasındaki etkileşimlerle inşa edilir.

Syntax: data tipi(class ismi) objismi=new counstractor();

OBJECT CLASS NEDİR?

Object class tüm diğer classların varsayilan olarak miras aldığı en üst sınıftır

VARİABLE NEDİR?

Programların calısmasi için gecici olarak bazi dataların saklanabileceği ve değistirilebileceği kutu.rezerved aria.

METHOD NEDİR?

Java'da "method" (metod), bir işlemi gerçekleştiren, belirli bir işlevi yerine getiren bir kod bloğudur.

Sizin oluşturduğunuzu sizin için is yapan aktif robotcuklar.bir kere yazilip ihtiyac duydukça cağırılarak kullanılır.

Syntax: accesmodifier +returntype+ method ismi+ methodparantezi+method bodysi

ASİGNMENT OPERATOR (ATAMA OPERATÖRÜ) VE DECLARATİON (TANIMLAMA)

Asignment Operator (Atama Operatörü) ve Declaration (Tanımlama), Java programlama dilinde önemli kavramlardır ve bir değişkenin değerini atamak ve tanımlamak için kullanılırlar. İşte bu kavramların açıklamaları:

1. **Assignment Operator (Atama Operatörü):** Atama operatörü, bir değişkene bir değer atamak için kullanılır. Java'da atama operatörü "=" işaretidir. Bu operatör, sağ tarafındaki değeri, sol tarafındaki değişkene atar.

**Declaration (Tanımlama):** Tanımlama, bir değişkenin veri tipini belirterek ve adını belirleyerek değişkeni oluşturma sürecidir. Java'da bir değişkeni kullanmadan önce, onu tanımlamak gerekir. Tanımlama, değişkenin veri tipini belirtmek ve ne tür değerler alacağını belirlemek için kullanılır.

JAVADA KAC TUR VE VERİ TİPİ VARDIR?

2 tür veri tipi vardır:

1 -primitive data types (ilkel) : 8 tane dir byte,sort,int, long,float,double,char,boolean,

2-non-primitive data types(ilkel olmayan, referans ): String, Array, Classlar…

PRİMİTİVE VE NON-PRİMİTİVE ARASİNDAKİ FARKLAR?(İLKEL VE REFERANS)

Primitive :sadece değer(variable) vardır,java tarafından oluşturulmuştur ve 8 tanedir,isimleri kucuk harfle baslar. Primitiveler stack memory de farklı boyutlarda yer kaplar.

Non-primitive: değerin yanında methodlarda vardır, javada oluşturabilir coder da oluşturabilir sinirsiz tanedir her class bir non-primitive data türüdür,buyuk harfle baslar. Non-primitiveler değisken boyuttadır.

DİZİN VERİ TİPİ NEDİR NASİL KULLANILIR?

Ayni türdeki birden cok veriyi tutmak için kullanilan non primitive veri türüdür.

Dizinler sadece primitive depolar.non primitivelerin referanslarını depolar.

Bir dizin olusturmak için tür ve boyutlar belirtilmelidir.

İnt[]arr=new int[5];

STRİNG VERİ TURU NEDİR NASİL KULLANILIR?

Metin verilerini temsil etmek için kullanılan veri tipidir. String nesnesi oluşturmak icin cift tırnak icinde metin kullanılır

String mystring=”hello word”;

İNT VE LONG ARASİNDAKİ FARK NEDİR?

Long veri tipi 8 byte kadar olan tam sayilari temsil eder daha iyi anlasılması için insan vucudundaki hucre sayisini belirtmek icin long data tipini kullaniriz

İnt veri tipi 4 byte kadar olan tamsayilar temsil ederken kullanılır. Daha anlasilir olması icin ülke nufusunu belirtmek için int data tipini kullanırız

Long data türü inte göre daha geniş kapsamlıdır ve long u kullanırken değerin sonuna ‘L’ karakteri yazmaliyiz

DOUBLE VE FLOAT ARASİNDAKİ FARK NEDİR?

Float memory de 4 byte yer kaplar ve ondalıklı sayilar icin kullanılır daha anlaşılır olması icin: fiyatlar ı bu data tipinde depolarız.değerin sonuna ‘F’ karakteri yazmaliyiz.

Double memory de 8 byte yer kaplar ondalikli sayilar icin kullanılır daha anlasılır olması icin hücre agirligini double data tipinde depolariz.

Double data type float a göre daha hassastir.

ASCİİ VALUES NEDİR?

Javada her karakterin sayisal bir değeri vardır bu degerler asci değerler olarak adlandırılır.bu değerler bulunduğu tablayo ise ascii tablo denir. Bir karakterin asci karsılığını bulduran code: int x=’A’;

JAVADA MEMORY KULLANIMI?

Javada 2 tür memory vardır stack memory ve heap memory

Stack memory kücük memorydir primitivleri ve non primitivelerin referanslarını depolarken

Heap memory büyük memorydir non-primitiveleri depolar.

EQUALS() METHODU VE == OPERATORU ARASİNDAKİ FARK?

Her ikisi de eşitlik karşılastırmak için kullanılır. equals methodu iki değiskenin sadece değerlerini kıyaslarken,== operatoru iki değiskeni değerinin yanında referanslarını da kıyaslar..biz iki stringi karşılastırırken bunların referansları ile ilgilenmeyiz sadece değerlerini kıyaslarız. Bu yüzden equals methodu kullanırız..== operatoru ise yine matematik işlemlerinde esittir in yerine kullanılır.

WRAPPER SINIFLAR NEDİR NEDEN KULLANİLİR?

Primitivelerde sadece deger vardır ve stack te depolanır ama bazen primitiveler içinde methoda ihtiyac duyulmuştur. Java bu ihtiyacı karsılamak icin wrapper class ı olusturmuşutur böylelikle primitive datalar non primitive cevrilmis yanina method eklenmis ve kapladığı alan genislediği icin heap memory de depolanmıstır.

AUTOBOXİNG UNBOXİNG NEDİR?

Pirimitive data tiplerinin non primitive cevrimle islemi autoboxing iken bunun tam tersi unboxingdir. Ornegin int veri tipinin Integer a donusmesi autoboxing iken , Integer veritipinin yeniden int e cevrilmesi unboxing dir.

Byte k=19; byte primitive k=k; -🡪 unboxing

char a=’T’; Character wrapperA=a; -🡪autoboxing;

TYPECASTİNG(TİP DÖNÜSTÜRME) NEDİR NASİL KULLANİLİR?

Numeric data typlarinin birbirine dönüstürülmesi

Kucuk data typelarin (ornegin byte) büyük data type cevrilmesi( ornegin int) autowedening: java bunu otomatik olarak yapar Autoweding(otomatik genişletme) . autoweding işleminde önemli olan memory gereksiz yere işgal edilmemeli..yani byte inte cevirmek yeterli iken longa çevirip memoryi bos yere doldurmamalıyız

Buyuk data typlarin( ornegin long) kucuk data type cevrilmesi(ornegin int) islemi ExpilicitNarrowing(acıktan daraltma).bu islemi yaparken java bunun dogrulugundan süphe eder ve uyari verir , biz bu islemi yaptığımızdan emin isek değerden önce parantez içine daraltmak istediğimiz data type yazmaliyiz expilicit narrowing isleminde data bozulur.

WRAPPER VE TYPECASTİNG ARASİNDAKİ FARK NEDİR?

Wrapper tür dönüştürmek yani primitive non-primitive veya tam tersi olarak çevirmek iken

typCastin tip dönüstürmektir yani numeric primitive data typelarin birbirine dönüştürülmesi

VALUEOF() METHODU NEDİR NE İÇİN KULLANİLİR?

Genelde tüm karakterleri rakam olan stringi numeric data typene dönüstürmek kullanılan method

İKİ DEGİSKENİN DEGERİ NASİL DİGİSTİRİLİR?

Bu işlem mümkündür swap islemidir. Gecici bir değisken oluşturarak bu islemi gerçeklestiririz bunun bir diğer yöntemi daha vardır aritmetic operatorler ile yapılan swap.

CONCATİNATİON ?

Yapistirmak birlestirmek matematikteki ‘+’ sembolü ile yapilir. Javada ‘+’ sembolunu iki sayi arasinda kullanırsak toplama islemi yapmis oluruz. ‘+’ sembolu iki string arasinda veya bir string bir sayi arasında kullanılırsa concatination yani birleştirme yapıstırma islemi yapar. Char data tipinde kullanılırsa ascii değerlerini toplar.

LOGİCCALL OPERATOR VE COMPERİSON OPERATOR LER NEDİR?

Logicall operator(mantik ) = or and || %%

Comperison operator(karsilastirma )<= , >= ,>,<, == ,!= (sonuc boolean verir)

POST-İNCREMENT PRE-İNCREMENT ARASİNDAKİ FARK?

İncrement artırma islemi toplama ve carpma islemi için kullanılır

Decrement azaltma islemi cıkarma ve bölme islemi için kullanılır

Pre increment ++x bu degiskenin degerini artırır sonra degiskenin yeni degerini döndürürken

Post increment x++ bu once degiskenin degerini döndürür sonra degerini artırır

ACCES MODİFİER NEDİR?

Acces modifier erisim düzenleyicidir..hangi islevi kimin kullnacağıını düzenler.

Public: bir proje içinde heryerden ulasilabilir

Protecdet: ayni packageden classlar ulasabilir başka packagelardaki child classlar ulasabilir

Default: (package private) sadece aynı package içinden erisim mümkündür

Private: sadece aynı class içinden erisim mümkündür.

LOOPLAR NEDİR NE İÇİN KULLANILIR?

Looplar tekrarlı islemler gerektiğinde kullanılır.4 tane loop vardır;

For loop; baslangic değeri calısma sartı ve increment/ dicrement degeri tek paranteze içine yazılır. Curli bress içine calışacak kodlar yazılır.

While loop başlangıc değere while döngüsü dişina yazılır,while parantezine calışma kosulu yazılır , curli brees {} içine calışıcak kodlar en altına da incerement dicrement yazılır.(önce calışma kosulu kontrol edilir sonra çalışır)

Do while loop önce calışacak kodlar yazılır sonra incerement/dicrement yazılır.son olara while parantezine çalışma kosulu yazılır.(en az bir kez kode çalışır sonra sartı kontrol eder.)

For each başlangic değeri, loop calısma sartı ve increment/dicrement kısmını kendisi halleder. Arrays ve collections lar için kullanılır

WHİLE-LOOP İLE DO-WHİLE LOOP FARKI ?

1) - while loop kullanırsanız java önce şartı kontrol eder sonra loop body içindeki kodu çalıştırır - do-while loop kullanırsanız java önce loop body içindeki kodu çalıştırır sonra şartı kontrol eder

2) - while loop kullanırsanız loop body içindeki kod bazen hiç çalışmayabilir - fakat do-while içindeki kod en az 1 kere çalışır

// eğer biz yazdığımız kodun en az bir kere çalışmasını istiyorsak do-while kullanmalıyız // yazdığımız kod hiç çalışmasada olur mühim olan condition diyorsak while loop kullanmalıyz

BERAK VE CONTİNUE YAPİLARİ ARASİNDAKİ FARK NEDİR?

* Break keywordu bir döngüyü kırmak için kullanılır
* Continue keywordu döngüyü atlatmak görmezden gelmek için kullanılır
* Break sistem calısırken bulunduğu döngüyü sonlandırırken
* Continue döngüyü sonlandırmaz sadece sıradaki diger seceneğe atlatır

SONSUZ DÖNGÜ NEDİR? (İNFİNİTİ)

Bir döngüde islevsellik karsılanmadığında yada sonlandırıcı kosul belirtilmediğinde program sonsuz döngüye girer ve memory dolana kadar sürekli calısır.

Bazi durumlarda sonsuz loop oluşturur ve loop u kırma şartını kendimiz belirleriz.

VARİABLE (DEGİSKEN) NEDİR? ÖZELLİKLERİ NELERDİR?

Programin calismasi icin gecici olarak bazi datalarin saklanabileceği kutu.rezerved aria,

* Değişkenler, değerleri saklamak için kullanılır ve veri türüne bağlı olarak belirli bir bellek alanında tutulurlar.
* Değişkenler, değer ataması yapılmadan önce başlangıç değeri olarak null veya sıfır gibi varsayılan değerlere sahip olabilirler.

Syntax: datatype+ variablename+asigment opt+variable degeri+; int age=13;

KAC TÜR VARİABLE VARDIR?

3 tür variable(degisken vardır)

1-instance variable: (object variable)

* main ve diger methodun dışında classın içinde olusturulur.
* Obje oluştururken olusturulur ve her objeye bu variable monte edilir obje ismi ile ulasilir.
* Değer atamasi yapılsada yapılmasada olur

2-local variable:

* main method veya olusturduğumuz baksa methodun içinde oluşturulan variablelere denir.
* Java local veriablelere değer atamasi yapmaz biz yapmak zorundayiz.

3-static variable: (class variable)

* Classa ait olmak demek.class üretildiği zaman üretilir.
* Tüm objeler icin ortaktır. Class ismiyle erisim mümkündür.
* Class içinde ve method dısında tanımlanır.

Declare: tanimlamak bildirmek int age ; age isminde memory de alan oluşturduk..declare ettik

İNİTİALİZE VE DEFAULT KAVRAMLARI NEDİR?

İntialize= baslatmak demektir. Bir variable ye bir değer ataması yapmaktir.

İnitialize deilmeyen variablelere java kendisi değer atamasi yapar. Bu degerlere default değer denir.

Numeric data type ler ve char için default deger = 0 iken

String data type için default değer =null

Boolean data type için ise false dir.

STATİC BLOCK NEDİR?

Bir app de methodlar calıstırılamadan önce bazi datalara ihtiyac duyulur. Bu datalar main methoddan önce eliminde var olmalıdır ve bunun için static blok kullanılır.

Static block bütün methodlardan (main method dahil) önce calısır. Bir classda birden fazla static blok varsa en üstteki önce calısır.

METHOD

Tabii, işte Java'da methodlarla ilgili olası soruların cevapları:

1. **Java'da method nedir ve nasıl tanımlanır?**
   * Method, bir sınıf içinde tanımlanan ve belirli bir işlevi yerine getiren kod bloğudur. Java'da methodlar, metot başlığı (method signature) ve metot gövdesi (method body) olmak üzere iki ana bölümden oluşur. Metot başlığı, metot adı, parametre listesi ve geri dönüş türünden oluşurken, metot gövdesi ise metotun içinde yer alan kod bloğudur.
2. **Method imzası (signature) nedir ve hangi bileşenlerden oluşur?**
3. Method imzası, bir metodu diğer metotlardan ayırt etmek için kullanılan bilgilerin toplamıdır. Bir method imzası, metotun adı, parametre listesi(method parantezi) .
4. **Java'da metotların geri dönüş türü (return type) nedir ve neden önemlidir?**
   * Metotların geri dönüş türü, metotun çağrıldığında döndüreceği değerin türünü belirler. Geri dönüş türü "void" ise, metot herhangi bir değer döndürmez. Diğer durumlarda ise, metot belirtilen türde bir değer döndürmelidir.
5. **Metotların kapsamı (scope) nedir ve nasıl belirlenir?**
   * Metotların kapsamı, metotun içinde tanımlanan değişkenlerin ve yapılan işlemlerin ne kadar süreyle geçerli olduğunu belirler. Java'da metotların kapsamı, metotun açılış süslü parantezi "{" ile başlar ve kapanış süslü parantezi "}" ile biter.

Scope ve bodyFark olarak şu şekilde özetlenebilir:

* Scope, bir değişkenin, nesnenin veya metodun erişilebilir olduğu bölgeyi belirtirken, body, bir kod bloğunun içindeki kodları ifade eder.
* Scope, değişkenlerin ve nesnelerin geçerlilik alanını belirlerken, body, o kod bloğunun çalışma mantığını tanımlar ve belirli işlemleri gerçekleştirir.

1. **Java'da metodun parametreleri nedir ve nasıl tanımlanır?**
   * Metodun parametreleri, metot imzasında belirtilen ve metotun çağrılması sırasında alınan değerlerdir. Parametreler, metot başlığı içinde parantez içinde belirtilir ve birden fazla parametre virgülle ayrılarak tanımlanabilir.
2. **Metotların aşırı yüklenmesi (overloading) nedir ve nasıl yapılır?**
   * Metotların aşırı yüklenmesi, aynı isimde birden fazla metotun bir sınıf içinde tanımlanmasıdır. Aşırı yüklenen metotlar, farklı parametre listelerine veya farklı parametre türlerine sahip olmalıdır.
3. **Metotların geçersiz kılınması (overriding) nedir ve nasıl yapılır?**
   * Metotların geçersiz kılınması, bir alt sınıfın üst sınıftaki bir metodu aynı isimde ve aynı parametre listesiyle yeniden tanımlamasıdır. Geçersiz kılınan metot, üst sınıftaki metodu geçersiz kılan alt sınıftaki metotun davranışını belirler.
4. **Java'da statik (static) metotlar nedir ve nasıl kullanılır?**
   * Statik metotlar, bir sınıfın örneği (instance) oluşturulmadan doğrudan sınıf adıyla çağrılabilen metotlardır. "static" anahtar kelimesi ile tanımlanırlar. Statik metotlar, sadece static değişkenleri kullanabilirler ve this anahtar kelimesiyle erişilemezler.
5. **Metotları nasıl çağırabilir ve kullanabilirsiniz?**
   * Metotları çağırmak için, metot adını ve uygun parametreleri belirterek metot çağrısı yapabilirsiniz. Metotları sınıfın içinde veya başka sınıflardan çağırabilirsiniz.
6. **Metotların işlevi (function) ve prosedür (procedure) arasındaki fark nedir?**
   * Java'da metotlar, işlev (function) olarak adlandırılır ve geri dönüş türü olan metotlardır. Yani bir değer döndürürler. Prosedür (procedure) ise geri dönüş türü "void" olan ve herhangi bir değer döndürmeyen metotlara verilen addır.
7. **Yerel değişkenlerle ilgili metotlar için ne gibi riskler olabilir?**
   * Yerel değişkenler, sadece metot içinde geçerli olan değişkenlerdir. Yerel değişkenlerin tanımlandığı blok dışına çıkıldığında, bellekten silinirler. Bu nedenle, yerel değişkenleri dikkatli bir şekilde kullanmak ve metot dışında erişilemez hale getirmek önemlidir.
8. **Metotları test ederken hangi teknikleri ve yaklaşımları kullanırsınız?**
   * Metotları test etmek için test senaryoları oluşturmak, farklı girdi değerleri ile metotları çağırmak ve beklenen çıktıları doğrulamak gibi test teknikleri ve yaklaşımları kullanılır. Ayrıca, JUnit gibi test çerçeveleri de kullanılabilir.
9. **Metotların geri dönüş türü olarak "void" kullanmanın anlamı nedir?**
   * Geri dönüş türü olarak "void" belirtilen metotlar, herhangi bir değer döndürmezler. Yani bu metotlar, çağrıldığında herhangi bir değeri geriye vermeyip, sadece belirtilen işlemleri gerçekleştirirler.
10. **Metotlarda exception handling (istisna işleme) nasıl yapılır?**
    * Metot içinde oluşabilecek hataların yönetilmesi için try-catch blokları kullanılır. Bu sayede oluşan istisnalar (exceptions) kontrol edilir ve gerektiğinde uygun bir şekilde işlenir.
11. **Recursion (özyineleme) kavramı nedir ve nasıl kullanılır?**
    * Recursion, bir metotun kendini doğrudan veya dolaylı olarak çağırmasıdır. Bir metot içinde recursion kullanarak tekrarlayan işlemleri basit ve anlaşılır bir şekilde çözebiliriz. Ancak, recursion kullanırken dikkatli olmak ve uygun bir base case (temel durum) belirlemek önemlidir, aksi takdirde sonsuz döngülere neden olabilir.

Parametre ve argüment nedir?

Methodu oluşturur parantez içine yazılan ve pass yapması için kullanılan veriable değerlere parametre denir

Main methodun içinde örneğin bir toplama methodunu cağırken ilsem yapılmasını istediğimiz değerlere argument denir.

CONSTRUCTOR

1. **Java'da constructor nedir ve ne işe yarar?**
   * Constructor, bir sınıfın nesnesi (instance) oluşturulduğunda çağrılan özel bir metottur. Nesne oluşturma işlemi sırasında sınıfın özelliklerine başlangıç değerleri atanmasını sağlar. Yani bir sınıfın constructor'ı, sınıfın nesnesi oluşturulurken otomatik olarak çağrılarak nesnenin başlangıç durumunu ayarlar.
2. **Bir sınıfın constructor'ı olmazsa ne olur?**
   * Eğer bir sınıfın herhangi bir constructor'ı tanımlanmamışsa, Java otomatik olarak parametresiz ve boş bir (default) constructor oluşturur. Bu default constructor, sınıfın nesnesi oluşturulduğunda çağrılır ve herhangi bir başlangıç değeri ataması yapmaz.
3. **Default constructor nedir ve ne zaman oluşturulur?**
   * Default constructor, parametresiz ve boş bir constructor'dır. Bir sınıfın hiçbir constructor'ı tanımlanmadığında veya sadece parametreli constructor'lar tanımlandığında, Java tarafından otomatik olarak oluşturulur.
4. **Bir sınıf içinde birden fazla constructor tanımlanabilir mi?**
   * Evet, bir sınıf içinde birden fazla constructor tanımlanabilir. Bu durumda farklı parametre listeleri ile constructor'lar oluşturulabilir ve farklı tipte nesnelerin oluşturulmasına olanak sağlanabilir (constructor overloading).
5. **Constructor'lar miras alınır mı?**
   * Constructor'lar miras alınmazlar. Alt sınıf, üst sınıfın constructor'larını direkt olarak kullanamaz. Ancak alt sınıfın constructor'ı çalıştırıldığında, otomatik olarak üst sınıfın default constructor'ı çağrılır. Alt sınıf, üst sınıfın parametreli constructor'larını çağırmak için "super" anahtar kelimesi kullanabilir.
6. **Constructor'lar ne zaman çağrılır?**
   * Constructor, bir sınıfın nesnesi (instance) oluşturulduğunda çağrılır. Nesne oluşturmak için "new" anahtar kelimesi kullanılır ve bu işlem sırasında ilgili sınıfın constructor'ı otomatik olarak çağrılır.
7. **Parametresiz bir constructor oluşturmanın avantajları nelerdir?**
   * Parametresiz bir constructor, nesne oluşturma sürecini basitleştirir ve başlangıç değerleri ataması yapmadan nesneyi oluşturmayı sağlar. Aynı zamanda parametresiz constructor, farklı senaryolara uygun olarak nesnelerin oluşturulmasını sağlar.
8. **Constructor'lar ve diğer metotlar arasındaki fark nedir?**
   * Constructor'lar, sınıfın nesnesi (instance) oluşturulurken çağrılırken, diğer metotlar isteğe bağlı olarak çağrılır. Constructor'lar geri dönüş türü (return type) içermezken, diğer metotlar bir geri dönüş değeri üretebilirler.
9. **Constructor'lar neden geri dönüş türü (return type) içermez?**
   * Constructor'lar, bir sınıfın nesnesini oluşturmak ve başlangıç değerlerini ayarlamak için kullanılır. Nesne oluşturulduğunda, constructor çağrılabilir ve geri dönüş değeri oluşturulan nesnedir. Bu nedenle constructor'lar geri dönüş türü içermez.
10. **Constructor'lar ve setter metodları arasındaki fark nedir?**
    * Constructor'lar, nesne oluştururken başlangıç değerleri atanmasını sağlar ve sınıfın özelliklerini başlangıç değerleriyle doldurur. Setter metodları ise varolan bir nesnenin özelliklerini değiştirmek veya güncellemek için kullanılır.
11. **Constructor chaining nedir ve nasıl yapılır?**
    * Constructor chaining, bir constructor'ın başka bir constructor'ı çağırmasıdır. Böylece, aynı işi tekrar tekrar yapmak yerine, bir constructor içinde yapılan işlem başka bir constructor'ı çağırarak gerçekleştirilebilir. Constructor chaining yapmak için "this" anahtar kelimesi kullanılır.
12. **Constructor'lar neden başka bir constructor'ı çağırır?**
    * Constructor'lar başka bir constructor'ı çağırarak kod tekrarını önlemek ve constructor chaining yöntemiyle nesne oluşturma işlemlerini kolaylaştırmak için kullanılır. Böylece, farklı parametre listeleriyle constructor'lar tanımlanabilir ve farklı senaryolara uygun nesne oluşturma işlemleri yapılabilir.

Method ve constractor arasındaki fark nedir?

* Method bizim olusturduğumuz ve bizim istediğimiz isi yapan cod blokları aktif robotcuklardır, constractor bir sinifın objesi olusturulurken cağrılan özel bir methoddur.
* Method a return type vardır, constractor de return type yoktur.
* Method her türlü ismi alabilir, constractor class ismini alır
* Method ismi kücük harfle başlar, constractor class ismi olduğu için büyük harf.

OOP

OOP NEDİR?

Oop object oriented programind yani obje tabanlı programlamadır

.komplex programlar yapmaya kucuk parcalardan başlanıp sonra bunlari birleştirerek sonuca ulasmaktır

Yani bir app oluşturacağımız zaman önce obje oluşturmaliyiz ve sonra bunları birleştirmeliyiz.

Daha anlaşılır olması için.söyle örnek verebilirim objleri lego parçası olarak varsayarsak bunları birleştirerek oluşturduğumuz araba ise app dur.

OOP NİN PRENSİPLERİ NELERDİR?

4 temel prensibi vardır.

1İNHERİTANCE(KALİTİM): iki sinifin arasında kurulan iliksiyi tanımlar. Yeni bir class oluşturmak istediğimizde daha önceki oluşturdugumuz classdaki özellikleri kapsiyor ise; yeni oluşturduğumuz class ı özelliklerini kullanmak istediğimiz class a child yaparız. Böylelikle iki class arasında kalıtım sağlamıs oluruz.

İnheritancenin avantajları:

* Code tekrarindan kurtulmamızı
* Code tamirini (maintenance) kolaylıkla yapabilmemizi
* Classımızın atomic olmasını sağlar

Java da bir classın sadece bir parenti olur buna ‘single parent’ denir. Coklu parente “multi parent” denir java multi parent yapısını desteklemez.

Ama parentin parenti onunda parenti olabilir bu yapıya ise “multilevel parent” denir.java bunu destekler

2 POLYMORPHİSM (COK ŞEKİLLİLİK):bir nesnenin birden cok biçimde davranma yetenegidir . kalitim yoluyla yapılır. 2 tür polimorfizim vardır.

OVERRİDİNG: gecersiz kılma yöntemi ile polimorfizim elde edebiliriz. Yani childe kullanmak istediğmiz method yoksa parent a gideriz fakat parent teki method cok daha kapsamlıdır. Bunu childe göre degistirebiliriz. Yani parenttaki methodu kullanırız ama body sini geçersiz kılar childe göre uyarlarız.

OVERLOADİNG: asiri yükleme yöntemi. Methodun isminin aynı kalip parametrelerinin farklı olması durumudur. Yani aynı methoda birden cok görev yükleyerek yapılan polimorfizm.

3 ENCAPSULATION(KAPSÜLLEME): veri saklama data hiding tir. Acces modifierin (erisim düzenleyici) private yapılarak dataya ulasımın kapatılması durumudur.

Ama istersek bu gizlediğimiz datayı getter methodu ile görüntülenmesine izin verir

Seter methodu ile ise değiştirilmesine müsaade etmis oluruz.

4 ABSTRACTİON(SOYUTLAMA):karmasik sistemleri daha basit ve anlaşılır kavramlara indirgeme islemidir . yani classların amacı bazen obje oluşturmanın dışında kategorizasyondur.bir claastan obje üretmeyeceksek sadece child classlarin yapacakları zorunlu göreveleri belirlemek için kullanacaksak abstraction yapıyor oluruz.

Bir class ı abstract keywordu ekleyerek abstract class yaparız

Bir methodu abstract keywordu ekleyerek abstract method yaparız ve body eklemeyiz.böyle yaparak bu methodu child classlarin kullanmasına mecbur bırakırız.

İnterface: java bir classın birden fazla parenti olmasına müsaade etmez. Ama bazı durumlarda böylesi yapılara ihtiyac duyulur.buna cözüm olarak java interface i oluşturmuştur. Yani bir class için coklu interface implement edebiliriz.

ABSTRACT VE İNTERFACENİN FARKI NEDİR?

* Abstact classlar hem abstract hem concreate(bodyli) method içerebilir.fakat interfaceler sadece abstract method içerir.(istisnalar haric)
* Abstract class “multiple inheritance” yi desteklemez.fakat interfaceler bunun için uygundur.
* Abstract classalrın içine her türlü veriable olusturulur.fakat interfacedeki veriableler public, static ve final olmak zorundadir.
* Abstract classın constructor ı vardır fakat obje üretilmez(cünkü abstract oldukları için eksik yapılardır.üretilen objede eksik olur.). İnterfacelerin ise constructor u yoktur bu yüzden obje üretilmez

THİS() VE SUPER() KEYWORDLERİ ARASINDAKİ FARK?

Sinif içindeki değiskenlerin, methodlarin veya counstructor larin farklı neslelerle olan iliksisini ifade eder.

This() ayni clastaki counstructor u secmek için kullanılırken

Super() parent classdaki counstructor u secmek için kullanılır.

Ve bir counstructor içinde ikisi aynı anda olmaz.

İNHERİTANCE TÜRLERİ NELERDİR?

3 tür inharitance vardır;

* 1 single inheritance : bir child in bir parenti olma durumu..(tam tersi multiple inharitance dir ve java bunu desteklemez)
* 2 multilevel inheritance: bir childin parenti olması , o parentin da parenti olması, o parentinda parenti olması…
* 3 hiyerarsik inheritance: buda bir parentin birden fazla childinin olmasıdir.

PASS BY VALUE VE PASS BY REFERANCE NEDİR?

PASS BY VALUE orijinal degerin değişmemesi demek.java methodların orijinal değeri değistirmesine müsaade etmez.methodlara değer yollarken orijinal değerin kopyasını oluşturup yollar, method kopya üzerinden islem yapar.böylelikle orijinal değer korunmus olur ..istersek yazacağımız kodlarla orijinal değeri de değiştirebiliriz.

PASS BY REFERANCE: methodlara değer yollanırken degerin referansları yollanır. Referans yollanınca method referanci kullanarak orijinal degere ulasır ve orijinal degeri değiştirir. Örnegin c# gibi pass by referance kullanan dillerde method veriablenin orijinal degerini değiştirir.

COLLECTİONS NEDİR? TEMEL BİLESENLERİ NELERDİR?

Collection aynı türden birçok seyi bir arada saklamaktır.

Javada collections ise ayni tür data tipine sahip olan bir çok nesneyi depolamak demektir. Ve collections non-primitive depolar.data yapisi yönünden ayrılan 7 tane collections vardır. Temel bilesenleri interfaceler ve classlardır

* ArrayList
* LinkeLlist
* Queue
* Dequeue
* HashSet
* LinkedHashSet
* TreeSet

ARRAY VE ARRAYLİSTİN FARKI?

array aynı tip coklu data depolar fakat ilkeldir eleman sayisini bastan belirtmek gerekir.ve collections değildir.array list ise yine aynı tip coklu data depolar fakat esnektir. Önceden eleman sayısını bildirmeye gerek yoktur.

ARRAYLİST VE LİNKEDLİST İN FARKI NEDİR?

Arraylist index ile işlem yapar bu sebeple search isleminde başarılıdır

Linkedlist pointer ile islem yapar ve eleman ekleme cıkarma isleminde başarılıdr

1 ARRAYLİST: ayni tip coklu datanın bir arada tutulmasidir. Esnektir. Bizim yüklediğimiz data kadar yer kaplar. İndex yapısı kullanır herhangi bir ekleme çıkarma islemi yaparken yeniden indexleme yapar ve bu cok zahmetlidir. Bu yüzden eleman ekleme cıkarma islemlerinde basarisizdir.

2LİNKEDLİST:arraylistlerin eleman ekleme ve cıkarma işlemlerindeki başarısızlığını java linkedlist ile çözmüştür. Linked listlerde index yoktur pointer vardır.ekleme cıkarma islemi yapılacağı zaman bu pointerı degiştirerek yapar.

Eleman ekleme cıkarma islemi yapacağımız zaman linkedlist tercih ederken

Elemanlar arasi search islemi yapacağımız zaman arraylisti tercih ederiz.

LİST SET VE MAP ARASİNDAKİ FARK NEDİR?

Listler aynı data tipinde tekrarlı eleman da saklayabilir

Setler ayni data tipinde unique eleman saklayabiliyor

Maplar farklı data tipi olan key ve value ciftlerini saklayabiliyor.

MAP NEDİR ?

Mapler collections değildir.collections aynı data tipinden birden fazla nesne depolarken mapler farklı data tipinden key ve value ciftlerini depolar. Key kısımları uniq dir value kısımları tekrarlı olabilir.

BİR MAP I SETE NEDEN VE NASIL CEVİRİRİZ?

Setler loplarla kullanilir fakat mapler loplarla kullanılmaz. Bunun için mapı önce sete cevirir sonra yapacağımız islemi yaparız.

Set<map.Entry<String Integer>> isim=mapismi.entrySet(0); syntax i ile mapi sete cevirebiliriz.

HASHMAP ,HASHTABLE VE TREEMAP ARASİNDAKİ FARK NEDİR?

* Hashmap=random sıralar hastable=random sıralar treemap= natural order a göre sıralar
* Hashmap =hizlidir hastable= hashmape göre yavastır treemap= en yavastır
* Hashmap=key e 1 tane valueye sınırsız null yazılmasına izin verir. HashTable=null a izin vermez. Treemap= key kısmında null a müsaade etmez valuede nulla izin verir
* Hashmap=thread-safe ve synron değildir. Hashtable=thread-safe ve synrondur. Treemap=thread-safe ve synron değildir

THREAD-SAFE VE SYNRONİZATİON NEDİR?

Thread-safe aynı anda çok is yapabilmek demektir. Çok is yapma durumunda ise sıralama yapmak cok önemlidir.hangisinin hangi sırada yapılacağı yani senkronizasyon gereklidir.

OVERWRİTE NEDİR?

Maplerdee key kısmı unique olurken value tekrarlı olabilir. Aynı key isminde ikinci bir entry eklemek istediğimizde hata almayız. Son yazdığımız entry önceki entrynin üzerine yazılır buna üzerine yazmak anlamına gelen overwrite denir.bu yöntem value güncellemek içinde kullanılır.

GARBACE COLLECTOR NEDİR?

Memory de adresi olmayan refere edilmeyen degerler için cop toplayicisidir. Garbace collector belirli aralıklarla memory i tarar ve adresi olmayan conteynırları önce finalize eder(yani silmeden önceki son rutuslar bir nevi paketleme) sonra siler.

FİNAL FİNALY VE FİNALİZE ARASİNDAKİ FARK NEDİR?

FİNAL: artık son anlamındadır.degeri değiştirilemez.yani sabittir...claslarla, methodlarla ve veriablelerle kullanılır.

* Veriablelerle kullanımı:initialize yapılmalı kimse bu değeri degistiremez.
* Methodlarda kullanımı:body değiştirilemez override edilemez
* Classlarla kullanımı: kısırlastirma durumu yani bu class parent olamaz.

FİNALY : herhalukarda calısan kodlar demektir. Bir classta exception olsa da olmasada çalısmasını istediğimiz kodu finaly block içerisine yazarız. Örnegin: data base ile baglantıyı kesme islemini yapan kodlar gibi.

Finaly block ne zaman çalışmaz: jvm aniden kapanırsa veya system.exit cağrısı durumunda çalısmaz.

FİNALİZE: memoryde garbace collector calısacağı zaman otomatik olarak finalize methodu ile silinecek degerler önce rutuslanır, gözden gecirmeler yapılır (yani bir nevi paketleme islemi) sonra silinir.finalize methodunu biz de cağırabiliriz, fakat bu tercih edilmez. Java bunu arka planda zaten yapar.

JAVADA Kİ HATA TÜRLERİNDEN BAHSEDER MİSİN?

Javada 2 tür hata vardır.

* 1 )exceptionlar(umulmadık hatalar) 2 tür exception vardır
  + 1 Run time exception
  + 2 compile time exception
* 2)errorlar(çözümü olmayan code ile düzeltilemeyen handal edilemeyen hata). Errorlar 3 tanedir
  + 1 stack memorynin dolmasında olusan error: Stack Ower Flow Error.
  + 2 heap memorynin dolmasında oluşan error: Out Of Memory Error.
  + 3 javadaki sanal makinanın bozulması : Vitrul Machine Error.

EXEPTİON NEDİR? ORNEK VEREBİLİR MİSİN?

Exception umulmadık hata demektir. 2 tür exception vardır,

1 program çalışınca ortaya cıkan exception- Run Time Exception-Unchecked Exception . bu exceptionlardan bazilari;

* Aritmetik Exception: matematiksel islem yaptığımızda alınacak exception türüdür.
* Number Format Exception: içinde rakamlar dışında character bulunan bir stringi numeric veri tipine çevirirken aldığımız exception.
* Array Of Bounds Exception: array de var olmayan index kullanıldığında alınan exception.
* Class Cast Exception: birbirine dönüştürülemeyecek data typlarini dönüştürmeye calıstığımızda alınan exception.

2 derleme anında ortaya cıkan exception- Compile Time Exception –Cheched Exception . bu exceptionlardan bazılari:

* File Not Exception : path doğru olup olmadığı durumunda cıkan exception
* IO Exception : in ve out hatalarında cıkan exception.

HANDLE NEDİR KAÇ ÇESİT HANDLE VARDIR?

Handle hata düzeltme islemidir.3 cesit handle yöntemi vardır;

* Try-catch : bu yöntem ile javaya sen dene bir ben yanındayım hata alırsan cözümün bu deriz.
* Throw: hatayi ilan eder geri çekiliriz.biz oluştururuz.
* Throws:bunu java ister.emin olmak ister hata ile karşılaşınca ne yapacağını bilmek ister.trows ile hata alırsan cağır ben gelirim deriz

TRY-CATCH BLOGU NASİL KULANILIR?

Try “dene” demek. Try block içine calısmasını istediğimiz codelari yazarız.

Catch block “yakala” demek. Catch block içine ise olası hata durumunda çalısmasını istediğimiz codeları yazarız.

CHECkED EXCEPTİON U NASIL HANDLE EDERİZ.?

Checked exception derleme anında alınan hatadır. Bunu handle etmemizin 3 yolu vardır:

* Method signature de throws keywordu ile beraber hata ismini belirterek önlem alırız.
* Try-catch blogu kurarak hata dturumunda yapılması gereken işlemleri yazarak
* Checked exceptionları uncheched exceptionlara (run time) dönüştürebiliriz.(try catch blogu kurma)

THROW VE TROWS ARASİNDAKİ FARK NEDİR?

İki keywordu de handle işlemlerinde kullanırız.

* Throw ile hatayı ilan edip geri çekilirken, throws keywordu ile olası hataya engel oluruz.
* Throw method bodysi içine yazılırken. Throws method signature dan hemen sonra method bodysinden hemen önce yazilir.
* Throw keywordu body içine istenilen yere istenildiği kadar yazılırken ; throws method parantezinden hemen sonra bir kere yazilir.
* Throw dan sonra new keywordu ve constructor yazılarak obje oluşturulurken; throws dan sonra exception class ismi yazılır.

EXCEPTİON UN HİYERARSİK YAPİSİ NASİLDİR?

Exception Class hiyerarsinin en üst sınıfıdır.bu siniftan türeyen unchecked (rte) hataları için; Aritmetik exception, null pointer exception… gibi alt exception classlar bulunurken

Checked exceptionlar için IO Exception alt class ı oluşturulmuştur

KULLANICI TANIMLİ (CUSTOM) EXCEPTİON NASİL OLUŞTURULUR?

Custon exception oluştururken exceptionumun isminde bir class oluşturum. Extence keywordu ile Exception Class a child yaparım ve class içinde counstructor oluştururum bu counstructor parenttekini cağırmalı ve msj verip vermeyecek şekilde oluşturabilirim.

ENUM NEDİR NASİL KULLANILIR?

Enum: degişme ihtimali olmayan dataların içinde depolandığı yapıdır.uygulamada ihtiyac duyulan bu değişmez datalar uygulamada depolanmalı cünkü app bağımsız olmalı, sürekli ihtiyac duyduğu dataları başka kaynaklardan almamalı kendi içerisinde tutmalı.

Enumdaki sabitlere ulaşmak için obje oluşturamayız cünkü enumlar otomatik olarak public, static ve final dir.ulasmak istediğimiz sabitlere; enumismi+ . koyarak ulasabiliriz..

ENUMLARA DEĞER ATAMASI NASİL YAPILIR?

Enum isminin yanına parantez içerisine değerleri yazılabilir.

ENUMLARDA COUNSTRACTOR VAR MIDIR?

Enumlarda counstructor vardır fakat obje oluşturmak için değildir. Görevi; enum içindeki veriableleri enum sabitlerine assayn etmektir. Ve veriableleri başlatmaktır.

İTARATOR NEDİR?

İtarator bir collection elemanları sırayla gezmek için kullanılan yapıdır.loopların yaptığı isi yapar. Sonsuz loop oluşma ihtimali yoktur performans farkı yoktur.

JAVA DA STRİNG STRİNG BULDER VE STRİNG BUFFER İ KISACA KIYASLAYARAK ANLATIR MISIN?

1. **String:**
   * **String**, Java'da metinleri temsil etmek için kullanılan bir sınıftır.
   * **Değişmezdir:** Bir kez oluşturulduktan sonra, içeriği değiştirilemez. Yani, üzerinde yapılan değişiklikler yeni bir **String** nesnesi oluşturur ve orijinal nesne değişmez kalır.
   * **Thread-safe'dir:** Çünkü değiştirilemez, birden fazla iş parçacığı tarafından paylaşılabilir ve güvenli bir şekilde kullanılabilir.
   * **Performans:** Eğer çok sayıda metin birleştirmesi yapılacaksa, **String**'in performansı düşük olabilir. Çünkü her bir birleştirme işlemi yeni bir **String** nesnesi oluşturur ve bu gereksiz bellek tüketir.
2. **StringBuilder:**
   * **StringBuilder**, değiştirilebilir metinler üzerinde çalışmak için kullanılan bir sınıftır.
   * **Değiştirilebilir:** Yani, üzerinde değişiklikler yapılabilir ve aynı nesne içinde metin güncellenebilir.
   * **Thread-safe değil:** **StringBuilder** sınıfı, senkronize edilmemiştir, bu nedenle birden fazla iş parçacığı arasında paylaşılması durumunda uygun senkronizasyon sağlanmalıdır.
   * **Performans:** Tek bir iş parçacığı kullanımında, **StringBuilder** metin birleştirme ve düzenleme için daha performanslıdır. Uzun metinleri birleştirirken, **String**'e göre daha az bellek tüketir.
3. **StringBuffer:**
   * **StringBuffer**, **StringBuilder** gibi değiştirilebilir metinler üzerinde çalışmak için kullanılan bir sınıftır.
   * **Değiştirilebilir:** Yani, üzerinde değişiklikler yapılabilir ve aynı nesne içinde metin güncellenebilir.
   * **Thread-safe'dir:** **StringBuffer**, senkronize edilmiştir, bu nedenle birden fazla iş parçacığı tarafından güvenli bir şekilde kullanılabilir.
   * **Performans:** **StringBuilder** ile benzer performansa sahiptir, ancak tek iş parçacığı kullanımında biraz daha yavaş olabilir çünkü senkronizasyon ekstra bir maliyettir.

**Hangi sınıfı kullanmalıyım?**

* Eğer metin üzerinde değişiklik yapmayacaksanız ve farklı iş parçacıkları arasında güvenli bir şekilde paylaşmanız gerekiyorsa **String** kullanın.
* Tek bir iş parçacığında metin birleştirme veya düzenleme işlemleri yapacaksanız, performans için **StringBuilder** tercih edin.
* Birden fazla iş parçacığı tarafından paylaşılacak ve senkronize bir şekilde güvenli kullanılacaksa **StringBuffer** kullanın.

Özetle, **String** değiştirilemezdir, **StringBuilder** performans açısından daha iyidir (tek iş parçacığı kullanımlarında) ve **StringBuffer** birden fazla iş parçacığı arasında güvenli paylaşım için kullanılabilir. Seçiminiz, ihtiyaçlarınıza ve senaryonuza bağlı olacaktır.