* S1: Java platformu neden bağımsızdır?

C1: Herhangi bir sistemde çalışabilen bayt kodlarından dolayı.

Aciklama: Java'nın platform bağımsız olabilmesinin nedeni **doğrudan makine diline derlenmemesidir**. Fakat bu Java'nın derlenmediği anlamına gelmez. Java kaynak kodları sadece Java Sanal Makinesinin-JSM (Java Virtual Machine-JVM) yorumlayabileceği bitkodlarına (bytecode) derlenirler. Bitkodları işlemci seçmezler.

Url: https://www.youtube.com/watch?v=CiOYIWVBTds

S2: Java neden %100 nesne yönelimli değil?

C2: Çünkü nesne olmayan 8 primitive data türünü kullanır.

Aciklama: Java, nesne olmayan sekiz ilkel veri türü (boolean, byte, char, int, float, double, long ve short) kullandığından% 100 nesne yönelimli değildir.

Url: : <https://web.cs.hacettepe.edu.tr/~bbm102/misc/java_notes_by_oa.pdf>

Url2: <https://bilisimprofesyonelleri.com/30-soruda-java-gelistirici-mulakati/>

S3: public static void main (String [] args) Java’da yer alan bu yapıyı açıklayınız.

C3: Bu, herhangi bir Java programı için giriş noktasıdır

public: bu yönteme kimin erişebileceğini belirtmek için kullanılan bir erişim

değiştiricidir ve bu yöntemin herhangi bir sınıf tarafından erişilebilir olacağı

anlamına gelir.

static: Java'da sınıf tabanlı olduğunu tanımlayan bir anahtar kelimedir. main (), bir

sınıfın oluşumunu oluşturmadan erişilebilmesi için Java’da static yapılır.

void: methodun return type dır.

main (): JVM tarafından aranan methodun adıdır.

JVM: Java sanal makinesi

String args ise main methoda iletilen parametredir.

Aciklama: Main metodu programin ana metodu olarak gecer. Cogu dilde bu boyledir. Main metodlarinin farkli sekillerde yazildigi farkli diller mevcuttur. Ana metodlar olmadan programin giris bolgeleri belirlenemez.

Hikaye ornegi verebilirim.

giris'i olmadan gelisme ve sonuc'u olan hikayeler olur mu? :)

Main metodunu her dosya icine eklemeyeceksiniz. Genelde main metodu ilk dosyanizda olur. Siz yeni siniflarinizi o metod altinda turetirsiniz. Ornegin

**public static void Main(String[] args)**

**{**

**Sinif snf = new Sinif();**

**snf.uyeIslev(Burada islemler);**

**}**

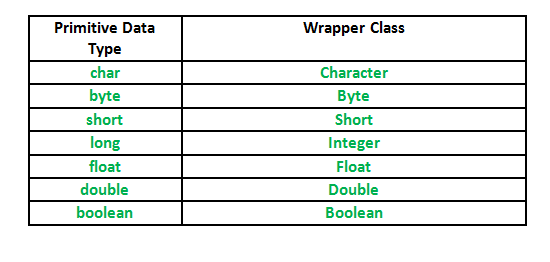
**Kolay gelsin**

JVM kodları çalıştırırken yazılmış onlarca class arasından nereden başlayacağını bilmesi gerekmektedir ve JVM standartlaştırılmış **public** **static** **void** **main(String[ ] args)**metodunu arar ve oradan çalışmaya başlar. İşte bu yüzden JVM e "sen ilk olarak bu metodu çalıştır" demek için bunu yapmalıyız.

S4: Wrapper Class’lar nelerdir?

C4: Wrapper Class’lar Java primitive datalarını o sınıfın başvuru türlerine veya o

Class’ın nesnelerine dönüştürürler.

Aciklama: 

Wrappler hakkında bilmemiz gerekenler şunlardır:

* Javada 8 tane primitive type vardır. Bunlara karşılık gelen sınıflara wrapper adı verilir.
* Wrapper’lar parametreleri String kabul edebilirler.
* Boolean wrapper, parametreye string olarak true verildiği sürece true döner, diğer durumlarda false döner.
* Character wrapper, ascii veya karakter alabilir.

Url: https://www.youtube.com/watch?v=L91h88SMOBw

S5: Java da Constructor nedir?

C5: Bir nesneyi başlatmak için kullanılan bir kod bloğunu ifade eder. Sınıfın adıyla

aynı ada sahip olmalıdır.

Aciklama: Oluşturulan sınıf yapılarının nesne olarak tanımlanması durumunda proje dosyasının alt yapısını hazırlayan, kurucu rol üstlenerek çeşitli ilk işlemleri gerçekleştiren, kullanılan sınıf yapısı ile aynı isme sahip olan, geriye değer döndürmeyen özel metot yapılarıdır.

Url: <https://www.youtube.com/watch?v=6UfXKx2Q59Q>

Url2: <https://www.youtube.com/watch?v=HKGsaeaXueM>

S6: Kaç çeşit Constructor vardır?

C6: 2 çeşit Constructor vardır.

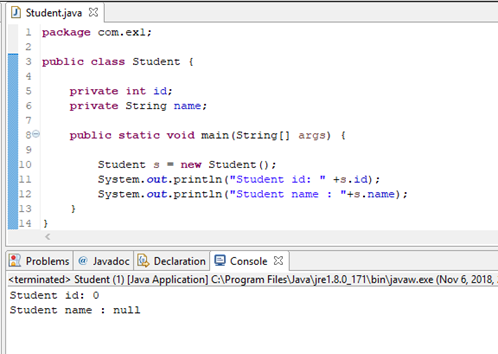
1) Default Constructor: herhangi bir girdi ya da değer almaz.

2) Parmeterized Constructor: durum değişkenlerini sağlanan değerlerle

başlatabilir.

## Aciklama: Varsayılan Yapıcı Nedir?

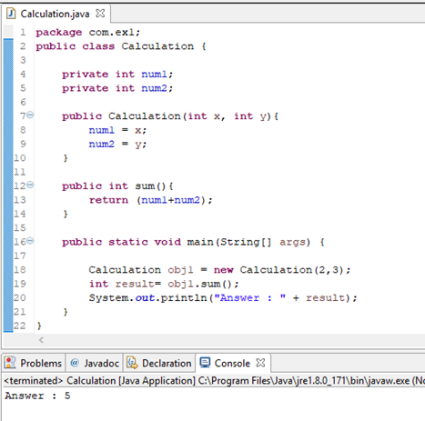
Yapıcı, bir nesne oluşturulduğunda çağrılır. Ayrıca o nesne için bellek ayırır. [Ayrıca, sınıftaki](https://pediaa.com/what-is-the-difference-between-class-and-structure/#Class) örnek değişkenlere başlangıç ​​değerlerinin verilmesine yardımcı olur . Programcı bir kurucu tanımlamazsa, program varsayılan kurucuyu otomatik olarak çağırır. Tüm üye değişkenleri sıfır veya boş olarak başlatır.



Parametreli Yapıcı Nedir?

Parametrelendirilmiş kurucu, parametreleri kabul eden bir kurucudur. Bir veya daha fazla parametre olabilir. Parametreli bir kurucu olduğunda, program varsayılan kurucuyu çağırmaz. Ayrıca programcı, yapıcı adından sonra parantez içindeki parametreleri bildirebilir.

Yukarıda Hesaplama adlı bir sınıf var. num1 ve num2 adında iki örnek değişkeni vardır. 7. satırda parametreli bir kurucu var. İki argüman x ve y alır ve bu değerleri num1 ve num2 örnek değişkenlerine atar.

Ayrıca sum diye bir yöntem var. Bu iki sayının toplamını döndürür. Ana yöntemde bir Hesaplama nesnesi vardır. Toplam yöntemi obj1 kullanılarak çağrılır. Son olarak, sonuçlar konsolda yazdırılır. 

S7: Bir Class i nasıl singleton yapabiliriz?

C7: O Class’ta ki Constructor i private yaparak.

Aciklama: **Singleton** tasarım kalıbının genel formu aşağıdaki gibidir. Bu tasarım kalının kullanımı oldukça basittir. **Singleton** uygulanacak sınıfın constructor medodu private **yapılır** ardından sınıfın içinde kendi türünden static **bir** sınıf tanımlanır. Tanımlanan bu sınıfa erişebilmek için **bir** metod sınıfa eklenir.

Url: https://www.youtube.com/watch?v=5DOEhu7LwKg

S8: Java da ArrayList ve Vector arasındaki fark nedir?

C8: ArrayList hızlı olması için senkronize edilmemiştir ancak Vector, iş parçacığı

açısından güvenli olduğundan yavaş olması için senkronize edilmiştir.

| **Karşılaştırma için temel** | **ArrayList** | **Vektör** |
| --- | --- | --- |
| Temel | ArrayList sınıfı Senkronize edilmedi. | Vektör sınıfı senkronize edildi. |
| Eski sınıf | ArrayList standart bir Koleksiyon sınıfıdır. | Vector, koleksiyon sınıfını desteklemek için yeniden tasarlanmış eski bir sınıftır. |
| Sınıf Beyanı | sınıf ArrayList | sınıf vektör |
| Yeniden tahsis | Belirtilmediğinde, bir ArrayList boyutunun yarısı kadar artırılır. | Belirtilmediğinde, bir vektör boyutunu iki katına çıkarmak için artırılır. |
| performans | ArrayList senkronize edilmediğinden, Vector öğesinden daha hızlı çalışır. | Vector senkronize edildiğinde ArrayList'ten daha yavaş çalışır. |
| Sayım / Iterator | ArrayList, ArrayList'te depolanan nesneleri dolaşmak için Iterator arabirimini kullanır. | Vector, Vektörler'de depolanan nesneleri geçiş yapmak için Numaralandırma ve Yineleyici arabirimini kullanır. |

S9: Java da equals () ve = = işareti arasındaki fark nedir?

C9: equals () operatörü primitive dataları ve Objectleri karşılaştırmak için kullanılır.

== ise iki tane Object i karşılaştırmak için kullanılır.

Aciklama: (==) ve Equals() methodları’nın ikisi de farklı 2 değeri karşılaştırmak için kullanılır. (==) operator’ü 2 nesneyi karşılaştırırken, Equals() methodu nesnenin içerdiği string’i karşılaştırır. Yani kısaca (==) operatörü 2 nesnenin referans değerlerini karşılaştırırken Equals() methodu sadece içeriği karşılaştırır.

Url: https://www.youtube.com/watch?v=jT06ibYdEXo

Q10: Java’da Heap ve Stack Memory arasındaki farklar nelerdir?

C10: Heap hafızası bir Application in tüm parçaları tarafından kullanılır. Objectler Heap

içinde depolanır ve herkesçe genel kullanıma müsaittir. Stack hafıza ise sadece bir

yürütme dizisi tarafından kullanılır ve diğerleri tarafından asla kullanılamaz.

Aciklama: Stack ve Heap Arasındaki Farklar

**Stack bellekten statik olarak yer tahsisi için kullanılırken, Heap dinamik olarak yer tahsisi etmeyi sağlar**. Her ikisi de Ram bölgesinde bulunur. Stack'te yer alan veriler direk bellek içine yerleştirilir dolayısıyla erişimi çok hızlıdır.

Heap ve Stack arasında ki en önemli farklardan birisi heap de veriler karışık bir şekilde saklanır stack de ise artan ya da azalan adres mantığında (big and little endian) çalışır. Buna bağlı olarak heap de ki bir veriye erişmek stack de ki bir veriye erişmeye göre daha maliyetli bir işlemdir. Başka bir farkı ise stack de ki veri hemen silinirken heap de ki veri Garbage Collector (Çöp Toplayıcı) algoritmasına bağlıdır.

Url: <https://www.youtube.com/watch?v=450maTzSIvA>

Url2: https://medium.com/t%C3%BCrkiye/stack-ve-heap-kavram%C4%B1-59adcb29d454