

Java. B n m I
 z eviri:  nci Gedik

A ş klama:

Bu , Java programlama hakkındaki bir seri makaleden ilkidir.  lk b n m Java'y  bir dil olarak , genel karakteristi i ve en fazla rabet g ren bilgi kaynaklar n yeriyile a ş klamaktad r.

Giri 

Bu ,Java programlama dilini ve onun  şevresel geli imini a ş klayan bir dizi makale ile ba lar.Birinci ama  olarak ,dilini tarihi hakkında yay mlardan, mikrodalgalar ve  sama r makinelerinden... uzak duraca m.(E er aran zda bunlardan herhangi biri ile ilgilenen varsa bunlar   şoklu k t phanelerden hatta literat rden bulabilir.) Ayn  zamanda kocaman burunlu,  şgen v cutlu ve ellerini marimba'n n ritmine g re sallayan ,bana eski Amerikan bilim kurgu filmlerini  hat rlatan Duke ' tan  da (Java'n n maskotu)  uzak duraca m. 

  Java fenomeni de zaman nda ge ici pop lerlik ta d .Her seviyedeki bilgisayar magazini bunun hakkında yazd  , Java apletlerini geli tirmek  zere  r n satan bir şok yaz l m firmas  var ve  şok az web sitesi Java aleti i ermiyor.

Bu seriler aras nda m mk n oldu  kadar ,  aslında temel i ler i sin basit fakat ustal kl  kullan m i sin hayl  kar  k ve zor olan bu programlama dilini derinlemesine incelemeye ve  renmeye  sal ca m. 

Bu ilk birkaç makale ile  amac m z  genel olarak dile bir tan mlama sa lamak b ylece daha sonra apletler,a y-ili kili planlar, JDBC,vb.  gibi konular n derinine inebiliriz.

 zunu belirtmeliyim ki Java genel

ama Ői   bir dildir, bunu yapmada oldukŐa pop 1/4ler olmas na raŐmen sadece web sayfalar na aplet yazmak iŐin kullan lmak 1/4zere yarat lmam t r. Java'n n sadece g r 1/4n 1/4  ne tak lamay z Ő 1/4nk 1/4 java apletlerinden Őok daha ilginŐ bir Őok alternatifi vard r.

Bu ilk makale tan mlamadan Őok pratiktir. Bir sonra ki makalede, dilin temel karakteristiklerini ve Őevresel geliŐimini aŐklayaca z. Sadece en sonda Java program n bir Őrne ini g rece iz.

Orijin

Java   Sun Microsystem Inc. 'de hayat n 18 ay n geŐirdi i yerde 1991'de   James Gosling, Patrick Naughton, Chis Warth, Ed Frank ve   Mike Sherindan taraf ndan oluŐturuldu. 1992 ve 1995 aras nda Bill Joy, Arthur Van Hoff, Jonathan Payne, Frank Yellia, Tim Lindolm ilk prototipin olgunlaŐmas  1/4zerine i birli i yapt lar.

Tarihlerden belli ki   Java WWW 'den  nce   ya da en az ndan onunla e zamanla   oluŐturulmuŐtur -- 1991'e kadar Tim Berners Lee HTML dilini geliŐtirmemiŐtir.

Java'n n yap s  C'ye oldukŐa benzemektedir ve obje a r 1/4kl  karakteristikleri de ayn  zamanda C++ 'a benzemektedir. Java   uygun ve mant k ŐerŐevesinde birbirini tutan bir programlama dilidir.

  C ve   C++ aras ndaki benzerlikler, Java'n n   C++ ' n internet iŐin bir versiyonu oldu u izlenimini verebilir, fakat  nemli teorik ve pratik farkla klar vard r. Java  zellikle C++ ile kar laŐr  lan obje k kenli parametreleri d zeltir ve ar t r.

Internet   Java'y  ortaya att  ve herkesin bildi i bir dil haline getirdi ve e zamanla   olarak Java, siberuzayda 1/4cretsiz edinilebilen objeler dizisini geniŐletti inden

Java: Java. B n m I

ber internette derin etki yaratt  (kendi  sal t r labilen programlar).

 ncelemek ve  ndirmek 

Java programlama i sin kesin bir  ok ara   vard r. Tesad fen hep  GNU/LINUX ortam n n  nerece yim (kelimenin tam anlam  ile kay plar n n bulunmas ):

Varolan ELF, 1.2.13  ve sonraki kabuklara ihtiya   duyan Linux tabanl  olarak yerine getirilmi  tir.

Ben JDK (Java Development Kit) ortam n n   , Netscape versiyon 2.0.14 ya da sonras  , ve   Xemacs gibi bir yaz  edit r  kullanaca m .Daima   Xwindows ortam nda  sal aca z . Herzaman gerekli olmamas na ra men , Apletleri ve programlar  bir grafik arabirimi ile geli  tirirken   buna ihtiya  m z olacak.

JDK da  t m  a  a  daki sayfadan indirilebilir :

<http://www.blackdown.org/java-linux.html>

Dosyalar  hemen indirmek yerine bu siteden ba  lant  aynalardan birini se  ebilirsiniz. Birka   ay  nce - son ziyaretimde - JDK-1.1.3 'nin son versiyonu (bu makale i sin kullan lan)   1.19 versiyonunun orada oldu  unu g rd m. Unutmay n ki her  eyi indirmeye kalkarsan z bu size 24 Mb'a m clolur. Oysa sadece ihtiya  n z olan  eyler yakla  k 12 Mb yer kaplamaktad r(deneyin!).

gunzip ile ar  ivleri a  t n zda ba  laman z i sin hi  bir engel kalmayacak.

Normal olarak da  t m n  /usr/local dizinine yapmak ,sonra   /JDK1.1.3 alt dizinini yaratmak ve beraberinde a  a  dakileri yapmak gerekir :

/doc   HTML format ndaki

Java: Java. B l m I

ofis d k manlar

/demo Java demo programlar

/lib K t phaneler

/bin JDK'nin belli arařlarının olduđu dizin.

  /lib dizininin iřinde "classes.zip" (dosyayı ařmayın) dosyasını buluruz , bu dizin JDK tarafından kullanılan derlenmiř sınıfları iřerir.

root dizini ile beraber "src.zip" dosyasını buluruz , bu dizin classes.zip iřerisinde yeralan kaynak dosyaları iřerir . Bunlar derlenemez ve sadece bilgi amařları da t mlarla beraber desteklenirler.

Ortamın Geliřtirilmesi
Daha  nce belirtildiđi gibi geliřme arařları /bin dizininde yeralır :
javac : java kaynaklarını
baytkodlarına ıeviren java derleyicisi.

Java : java yorumlayıcısı
. Java baytkodlarını ısalıřtırır.

jre : Java'ya benzer ba ka bir yorumlayıcı , fakat kullanıcılar iřin olduđu d lmesinin ra men t m seřeneklerin kullanılabilir olması istenmemektedir.

appletviewer:   apletleri test eder ve ısalıřtırır.

jdb: hata d zelticisi.

Java: Java. B n m I

javap: Java da derlenen baytkod dosyalar n n ayarlar s .

javadoc : d k mantasyon jenerat r , genel ve korunmu  s n flar , aray zleri , yaplar , methodlar ve alanlar tanmlayan HTML sayfalar n  yaratır. Aynı zamanda s n f hiyera isini ve yelerin i inde bulundu u indeksi yaratır.

javah : Java programlar n n bilinen metodlar na eklenen ara  (C de)

jar: java s n flar n  ve kaynaklar n  bir Jar ar iv dosyasında saklar .

javakey : Dijital imzalar  koruyan ara .

updateAWT : bir ba vurudaki AWT1.02 metodu ile de tirilen isimleri g ncelle tirir.

Belli bir dizin yap s  ile glitch' ler olmadan  sal rken /usr/local/JDK1.1.3/bin 'i PATH de tirilmesine ataman z  tavsiye ederim .E er kullanılabilen sistem geli me  evresini daha geni  yapmak istiyorsanız /etc/profile dosyas  i indeki Path de tirilmesini kar t r n.

Karakteristikler

Devam etmeden  nce , kendimize Java nedir sorusunu sormam z gerekir. Cevaplamak i in JDK'n n elkitabında yazd  gibi Java'n n bir programlama dili ve aynı zamanda bir platform oldu unu g z nde buldururuz.

Bir programlama dili olarak Java a a daki karakteristiklerle beraber y ksek seviyeli bir dildir. (aynı noktada onlar  detaylı olarak inceleyebiliriz):

- Basittir
- Objeler kenlidir
- Daireler ve sınıflar
- Yorumlanabilir
- Kuvvetlidir
- Güvenilir
- Sistem yapısında taraftadır
- Taahhüt edilebilir
- Çok okuyuculu
- Dinamik
- Yüksek seviyeli

Dikkatimizi  şeken ilk  yey Java'nın yorumlanabilir olmasıdır. Java kaynakları bir objeyi baytkodlar olarak yaratması i sin bir  nceki daha d k seviyede ki derleyiciye ihtiyacı olmasına rağmen , de yici platformlarda yorumlanabilir .

" Birkez yazıp her zaman  şalattırabildi yimiz " Java baytkodlarına te ekk r ederiz. Java programları bir platformda derleyebilir ve JVM (Java Virtual Machine) aracını i şeren tamamen farklı ba ka bir yapıda  şalattırabiliriz.  rneğin bir Java programı Windows NT platformunda derlenebilir ve bir Sun Ultra Solaris 'de sorun olmadan  şalattırabilir . (en azından teorik olarak)

Genel olarak bir platform uygulamasını  şalattırma bir donanım/yazılım ortamıdır . Bununla beraber Java i sin bir platform  şalattırma bir yazılım ortamıdır . Her Java platformu bir donanım ortamında  şalattırılmak zorundadır.

Bir Java platformunun iki b l mü vardır :

 

-   Java Virtual Machine (JVM)
- Uygulamalar program aray z (API)

JVM , Java  ncelikli derlenen programlar n  sal t r ld y  a k bir bilgisayard r. K k ve basit olarak dizayn edilmi tir  nk d n len onu her yere indirmektir . Bu basitlik onu t m varolan platformlara ta may  m nk n k l yor , b ylece hepsine ortak bir yaz lm sa l yor ve her kar  k a  sisteminde ( nternet gibi) b y k  l de ta nabilirlik sorununu ortadan kald r r. Hayal edilebildi i gibi bu, durdurulamayan teknoloji i in yakla k bir nedendir. Bununla beraber sanal makine bir yaz lm tabakas  olarak g r lmemelidir, fakat isminin belirtti i gibi, bu dolu bir bilgisayar sistemi olmaya  sal y yor, yava  yava  ger k Java makinelerini , kredi kartlar  , TV dekodeerlerini ve benzerlerini destekleyen Java  tiplerini marketlerde g r yoruz .  

Java'n n API 's n grafiker kullan c  aray z (GUI) gibi  zellikleri destekleyen geni  bir yaz lm bile enleri koleksiyonudur . Bunlar k t phanelerde grupland r lm t r. (Java terminolojisini kullanan   Java paketleri). Bu paketler sayesinde  ok bilinen   web Apletlerinin yan s ra,   Java'y  daha bir  ok uygulamada kullanmak m nk nd r. Web serverlar n  dizayn etmek m nk nd r,  rne in, vekiller, ileti serverlar , IRC serverlar , ve  ntenet ile ilgili hayal edilebilecek herhangi bir  eyler...

 

- API paketlerini de yi ken gruplar i erisinde   s n fland rabiliriz :

Temel: objeler, matrisler, sayılar,
girdi ve   kt  , bilgi yapıları, sistemin
 zellikleri, tarih, zaman, vb.

- Apletler : fonksiyon seti ve Java aleti yapmak i in kullanılan  zellikler.
- A   sal mas :
URL'ler, TCP, UDP, soketler, IP adresleme.
- Uluslararasıla tma:
yerel tanımlamalara kolayca adapte olabilen kodlar yazmaya  yardım eder ve yerel dilde g r nt lenir.
- G venlik: d k ve y ksek seviye, dijital imza, i erir, genel ve  zel anahtarlar, kontrol ve sertifikaya giri .
- Yazılım B l mleri:
JavaBeans olarak bilinir.
- Veritabanları:  ok sayıda ili kili veritabanına giri i sa lar .
 
 

Bu temel API kabu unun yanıs ra
ayn  zamanda 3D uzantılar , ileti imler, animasyonlar,
vb. vardır.

B l m  n ba ında
Java'nın detaylı  sal abilece imiz
bir dizi karakteristi i oldu unu belirttik : 

 

Basittir: 

Java'da  daima bir tasla  olu turan bir  ok  iyi tanınan m  formular vardır. Bu , g  

- Aret  si aritmeti  i
- Kay teder (struct)
- Tip tan mlamalar (typedef)
- Makrolar (#define)
- Belle  e ihtiya   duyar (free)
-  oklu miras yok
- Ynlendirici operatrlr yok
-  enitelerde de  i  im yok

Obje kkenlidir:

Java  siziklerden dizayn edilmi  tir, ve sonu   olarak obje kkenlili  e temiz , yararlı ve pragmatik

Da  t  m  t  r:

Java d  ncede geni   TCP/IP birle  tiricisi ile dizayn edilmi  tir. Aslında , bu geli  tiricili

Kuvvetlidir:

Java g   yaz  m   bir dildir, bu derleme zamanında tip kontrol  ne izin verir .  tal  

Yap  sal ba  ms zd  r :  

Java'y   dizayn edenlerin as  l amac   "birkez yaz, her zaman ve daima herhangi bir yerde  sal  t.

G  venilir:

Da  t  m  t  r bilgiye ihtiya   m    teri operatrlr sisteminden en y  ksek seviyeli g  venli  

- Bir baytkodu ara  t  r  c  s  
-  tal  ma zamanında haf  zan  n de  erlendirilmesi
- Dosya giri   k  s  tlamas  

Java: Java. B n 1/4 m I

Derleyicinin sadece doÄ ru kodu oluÅ turmas na raÄ y men, yorumlay c   onu   derlemeden b

TaÄ nabilir:

Java'n n taÄ nabilirliÄ i bir yana, yap sal baÄ msizliÄ na g re, Java taÄ nabilir b

Yorumlanm t r:

Java'n n temel amaŐlar ndan birini yerine getirmek iŐsin, platform , java derleyicisi orta d zeyli b

 ok okuyucu:

SaÄ lam bir yolda birŐok taslaÄ  ayn  zamanda beraber  sal t racak Java uygulamalar 

Dinamik:

Java  sal ma zaman na kadar uygulamalar  oluÅ turan t m mod lleri   birbirine baÄ lama

Bu uzun "konu ma" dan sonra ( so u
Java d k manlar nda bulunabilir) Java'n n GNU/Linux
d nyas ndaki yerini merak edebilirsiniz .Daha  nce
Linux iŐsin kullan lan,   Sun Mikrosistem taraf ndan
geliÅ tirilen ticari amaŐsuz ( cretsiz daÄ t ld Ä 
halde) JDK d k manlar ndan bahsetmiÅ tik.

GLP lisans  altında

[GUAVAC](#) derleyicisi

gibi herhangi bir Java kaynaÄ n  hiŐbir sorun
ile karŐlaŐmadan derlememizi saÄ layan birŐok
araŐ vard r. Ayn  zamanda Berkeley lisans  altında  
[KAFFE](#) olarak adland r lan

Java: Java. B n 1/4 m I

bir sanal makineden bahsetmeliyiz .Son iki proje , geli me sistemini tamamlamak i nin Sun'n n Java s n f k t phanelerine ( cretsiz da t mda olanlar) ihtiya lar  oldu u halde olduk a iyi seviyedelerdir. 

Varolan bir ok proje, h cl  ilkel seviyede olmalar na ra men , geli me ortam n  h zli  ve sanal uygulamalar ile sa lamay  ama iyor.

Son olarak nette Java ve GNU  teknolojisinde  b y k hareketlenmelerin oldu unu  ,  zellikle Sun'dan ba ms z olarak Java i nin  cretsiz da t lan tamamlanmi  ara  kutular n n yarat ld n  sevinerek s yli yorum . Java'n n  neminin modas  ge ti inde yeni yeni anl yor olabiliriz, ama eminim ki onu hat rlatacak baz   yeyler olacakt r.(Hatta bir ok  eyin kalaca na bahse girerim)

JAVA da ilk program

Dilin genel karakteristiklerinden baz lar n  g rd kten sonra ger ek bir programa bakal m.Giri te belirtti im gibi, merhaba d nya  rne i gibi s radan laflardan m mk n oldu u kadar uzak duraca m, onun yerine istedi imiz boyutta bir kare  izecek olan bir program ile ba layal m.

(Fazla olmamak kayd  ile biraz kar  k): 

[File](#)
[Cuadro.Java](#)

```
class Cuadro {
public static void main (String args[]) {
       int l ,a;
      if (args.length == 2) {
      1 = Integer.valueOf(args[0]).intValue();
      a = Integer.valueOf(args[1]).intValue();
      }
      else {
      l= 20;
      a= 15;
      }
}
```

```
for (int i=1; i>0; i--){
    System.out.print("*");
}
System.out.print("
");
for (int i= a-2; i>0; i--){
    System.out.print("*");
    for(int j=1-2; j>0; j--) {
        System.out.print(" ");
    }
    System.out.print("*
");
}
for (int i=1; i>0; i--){
    System.out.print("*");
}
System.out.print("
");
}
```

Java'da kaynak dosyaya verilen isim ** şok** ** nemlidir**  nk  bir "derleyici birim" tan mlar. Dosya bir veya daha fazla tan mlama s n f  i şerebilir. Derleyici kaynak dosyadan uzant y  ta y mas n  ister. Java (4 karakter) kesinlikle baz  sistemler taraf ndan desteklenmemektedir. (DOS ya da Windows 3.1)

Verilen  rnekte s n f n ismi kaynak dosyanın ad  ile tan mlan r .Bu tesad f de yildir , Java'da t m kodlar bir s n fta olmal dır. Anla maya g re s n f n ismi ,s n f  i şeren kaynak dosyanın ad  ile uyuy mak zorundadır. Bundan ba ka Java k ş k b y k harfe kar y  hassast r.

Komut sat r nda  rne yi derlemek i şin :

› javac Cuadro.java

Java derleyicisi , (hersey iyi giderse) kendili yinden ikilik (baytkod) formu i şeren Cuadro.class dosyas n  yarat r ve acil olarak yorumlay c  taraf ndan  şal t r labilir :

› java Cuadro

Java kaynak kodu derlendiğinde , her sınıf arşivde sınıf ismi ile ayrı ve bireysel olarak yerleştirilir. Bu yüzden kaynak dosyalara işerdikleri sınıf ismi ile ayrı ad vermek güzel bir alışkanlıktır böylece kaynak dosya adı arşiv.class dosyası ile ayrıdır.

Cuadro.java örneğimizin basitliğine rağmen Java'nın temelini ve tipik java programların yapısını anlamamıza yardımcı olur .

Ölk olarak bu örneğin HTML dosyasında varolan bir applet olmadıkça fakat komut satırında java derleyicisi tarafından derlenebilen basit bir program olduğunu anlayın.

Programın ilk satırda:

```
class Cuadro {  
Ölk kelime saklanan dilden alınmış  
 , "Cuadro" -- ispanyolca bir kelime-- isimli yeni bir sınıfın  
tanımlamasını yapmaktadır. Tüm yelerini  
işeren sınıf verilmiş tanımlaması  
iki köşme parantezinin {} içinde verilir.
```

Bir sonraki satırda:

```
public static void main (String args[]) {  
main() isimli metod oluşturulmuş  
Java uygulamaların main() olarak isimlendirilen metod ile  
işalâna bayağılan  
(C/C++'a benzer şekilde). Şimdi size örneği anlamak  
işin gerekli detayları belirtmem izin verin .(sonraki makalelerde  
bu daha ayrıntılı olarak açıklanacaktır).
```

```
public ifre kelimesi sınıfı  
ye metodunu kontrol eder . Bir sınıf public olarak  
tanımlandığında , bu sınıf kendisinin  
dâiminde tanımlanmış kodlar tarafından  
kullanılabilir . static ifre kelimesi main() metodunu  
sınıfı araştırmadan uyanmaya zorlar .
```

Java: Java. B n 1/4 m I

Void bu metodun herhangi bir deÄere d n 1/4  mediÄiniÄ g sterir . Bir fonksiyon ya da metodun parametrelerini ge mek i in fonksiyon isminden sonra parantezler kullan lar , bizim  rneÄimizde main String s n f ndan indisleri olan bir matrisi parametre olarak al r .

Belli ki bu metoda baÄli t m kodlar ayn  zamanda yuvarlak parantezler aras nda kapat lar . Sonra, sat rda :

```
Ä int l, a ;
```

iki deÄiÄken tan mlar z

.Java  da t m degiskenler  kullan madan  nce  tan mlanmal d r .  zunu farkedin ki koddaki t m yap lar noktal  virg l ";" ile biter . 

Kodun kalanlar  aÄy daki gibi k  k bir algoritma g sterir :

- Arg man n saÄ taraf ndaki numaran n saÄlan p saÄlanmad Ä n  test eder.

```
If (args.length==2)
```

- Parametreler saÄlan rsa,tamsay  olarak atan rlar :

```
Ä l = Integer.valueOf(args[0]).intValue();  
Ä a = Integer.valueOf(args[1]).intValue();
```

- DiÄer durumda deÄiÄkenler varolan deÄerlere ilk olarak atan r :

```
Ä l = 20;  
Ä a = 15;
```

- Program n geri kalan k sm  ekranda kare  zmeyi saÄlayan yap dan baÄka birÄey deÄildir.  zel olarak print() 'in arg man olarak ge en diziyi bast Ä  yerde :`System.out.print()` ifadesi  ile g sterilebilir. System  sisteme giriÄi saÄlayan s n f n  ntan mlamas d r ve  out konsol ile baÄlant l    kt  dizisidir.Ayn  zamanda,`for (int i=1; i>0; i--)` BeklendiÄi gibi  C/C++ deneyi ve diÄer dillerin dediÄi gibi hareket eder.  devin iyi bir par sas  da okuyucunun bu k  k kodu kullanarak bunu nas  derleyip,  sal rabileceÄini test edebilmesidir, fakat ayn  zamanda  Linux, windows 95 gibi deÄiÄik platformlar alt nda  Quadro.class  derlenen objesini  sal rarak taÄ nabilir  kodu da test edebilirler ve herhangi bir platformda ayn   ekilde  sal r Ä n 

g rebilirler.(bunu genele almay n)

 zet

Java programlama dilinin genel  zellikleri ile ilgilendik.Programa hen z ba lad k ve gidecek uzun bir yol var,gelecek makaleler dilin tan mlamas  hakk nda daha derinlemesine bilgi verecekler.Bundan sonraki makalede , tan mlamalar ve de i kenlerin tipleri, temel yap lar, s n flar gibi dilin ana  zellikleri  zerinde  sal yaca z.

Herbir makalede o makalelerin referanslar n ,k t phanelerini ve URL'lerini verece im.

- JAVA Reference Manual. Patrick Naughton Herbert Schildt. McGraw-Hill. 
- Programming with JAVA. Tim Ritchey. Prentice Hall. 
- Tutorial de Java . Spanish manual in HTML format that can be obtained at www.fie.us.es/info/internet/JAVA. Agust n Froufe. 
- The Java Tutorial. English manual in HTML www.javasoft.com. Sun. 
- Tutorial from the JDK distribution. 

Michel Luvada  ngilizce'ye  sevrilmi tir.

Yazar: Erkan Kaplan

Son G ncelleme: 2005-03-08 18:46