

GARBAGE COLLECTOR (ÇÖP TOPLAYICI)

Merhaba arkadaşlar, bu yazı Java' da çöp toplayıcı kavramını anlatmak için yazılmıştır. Umarım yararlı olur.

ÇÖP TOPLAYICI NEDİR?

Öncelikle " Çöp toplayıcı nedir? " sorusuyla başlayalım; Programda bazı nesnelerin kullanımı bittiğinde bellekten silinmeleri gerekir, nesneye yönelik programlama dillerinde bu işlem otomatik olarak gerçekleşir. Yani programcının müdahalesi gerekmeden hafıza yönetimi gerçekleşir. İşte bunu gerçekleştiren yapı ise çöp toplayıcıdır. Çöp toplayıcısının temel görevi, kullanılmayan nesnelere bulmak ve hafızadan silmektir.

-
- Ancak çöp toplayıcısının ne zaman çöpleri temizleyeceği belli değildir !
 - Java belirli aralıklarla programı kontrol edip , işi bitmiş ve işaretlenmemiş nesnelere siler. Bu şekilde biriken nesnelere programda **OutOfMemoryException** hatası verir.
 - Eğer bir nesne **new** anahtar kelimesi ile oluşturup belleğe yerleşmek ister ve bellekte de yer kalmamış olursa; çöp toplayıcı hemen bellekteki gereksiz nesnelere silerek yer ayırır.
 - Nesnelere, belleğin heap alanında toplandığı için çöp toplayıcı bu alanda işlem yapar.
 - Çöp toplama isteğini **JVM** (Java Virtual Machine) gönderir.□
 - Kullanılmayan nesnelere temizleyerek o alanı belleğe tekrar iade eden çöp toplayıcı otomatik olarak çalışır.

Çöp Toplayıcı Hakkında Bilinmesi Gerekenler

- Çöp toplayıcının temel görevi , kullanılmayan nesnelere bulmak ve hafızadan silmektir. Ancak çöp toplayıcının ne zaman çöpleri temizleyeceği belli değildir. Bu belirsizlik nedeniyle bazı durumlarda çöp toplama aracını biz programcılar olarak çalıştırmak isteyebiliriz. □
- Bu araç tetiklense bile, özellikle tetiklenen anda çalışması garanti değildir. Çünkü bu aracın çalışması için bazı durumların olmuş olması gerekiyor. Bu durumların en önemlisi ve ilki ; sanal makine için ayrılan hafızanın azalmış olması gerekiyor. İkinci durum ise kullanılmayan nesnelere çoğalmış olması gerekiyor. Bu iki durumun oluşması çöp toplama sistemini otomatik devreye sokacaktır.

Çöp Toplayıcı İşleyişi

- Çöp toplama yöntemi çeşitlidir. Eski yöntem ve yeni yöntem olarak 2 gruba ayrılabilir.
- Eski yöntem referans sayma yöntemiyle yapılırdı. Bu yöntemde, bir nesne oluşturulduğunda kendisine ait bir sayaç çalışmaya başlardı. İlgili nesne her kullanıldığında bu sayaç arttırılır, nesnenin yoğun kullanıldığını temsil ederdi. Ve nesneyi kullanan diğer parçalar nesneyle ilişkisini kestiğinde, bu sayaç değeri azalırdı. Bu sayaç değeri ne zaman sıfırlanırsa çöp toplama sistemi tarafından hedef seçilirdi. Çünkü sayacı sıfırlanan nesne kullanılmayan nesnedir.
- İkinci yöntem ise JVM içinde HOTSPOT (sıcak bölge) teknolojisi vardır. Bu teknoloji sayesinde hafızadaki nesnelere eski ve yeni grup olarak ikiye ayrılır. Şayet bir nesne zaman içerisinde bir çöp toplayıcı tarafından temizlenmişse , hafıza bölgesindeki eski grup alanına kopyalanır. Çöp toplama aracı genel olarak eski

nesnelerin bulunduğu bölümle ilgilendiği için , eski grup alanına kopyalanan çöp nesne temizlenir.

- Ayrıca çöp toplayıcı referansa bağlı olmayan nesnelere hafızadan siler.
- `new Personel(); //referans yoktur` □
- Ancak bir nesne referansa bağlı ise çöp toplayıcı bunlara dokunamaz!

System.gc() Komutu

Bazı durumlarda çöp toplayıcıyı çağırmanız gerekebilir. Bu durumda `System.gc()` komut ile istediğimiz zaman çöp toplama aracını tetikleyebiliriz. Tekrar belirtmek gerekirse, bu komut bile çöp toplayıcısının hemen çalışacağını garanti etmez. Ayrıca bu komut yerine `Runtime.getRuntime.gc()` komutunu da kullanabiliriz. Aşağıda bu konuyla ilgili örnek oluşturulmuştur. Örneğimizde `Personel` adında bir sınıfımız vardır ve bu sınıftan üretilen iki adet nesnemiz vardır. Aynı örnek üzerinden değerlendirme tekrar yapılacaktır.

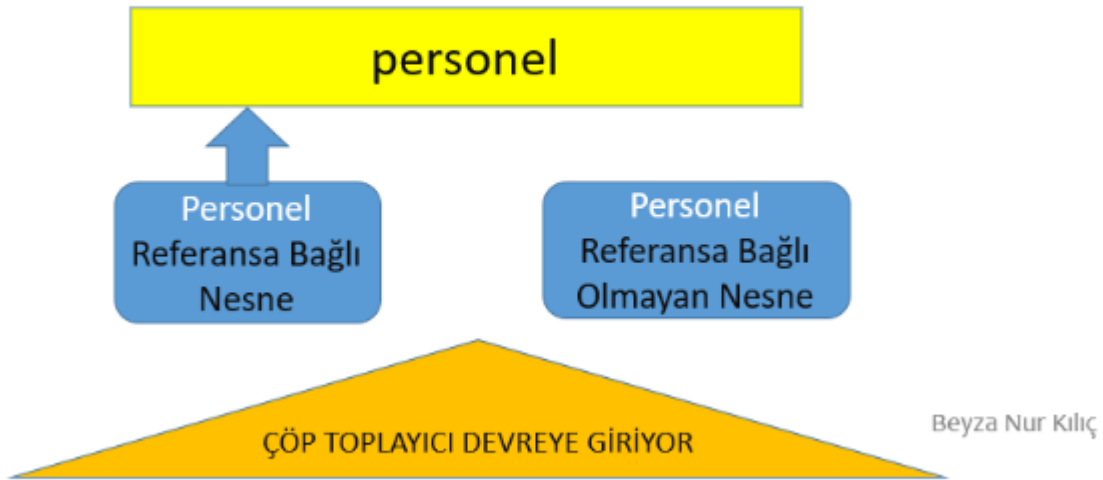
Örnek:

```
public static void main( String[] args) {  
  
    Personel personel = new Personel();    //nesne oluşturuldu  
  
    new Personel();  
    //isimsiz bir nesne oluşturuldu  
  
    System.gc();                            //çöp  
    toplama aracı tetiklendi  
  
}
```

Örnek Değerlendirme

- İlk oluşturduğumuz personel nesnesi hafızada yer alacak, kullanılmadığı takdirde hafızadan silinecektir.
- İkinci oluşturduğumuz personel nesnesi ise isimsizdir yani bir referansa bağlı değildir. Dolayısıyla çöp toplayıcının ilk hedeflerindedir çöp toplayıcı bunu derhal yok edecektir.

Örnek Şekillendirme



Çöp toplayıcı devreye girdikten sonra:



Finalize Metodu

- Biz javada çöp toplayıcının çalışmaya başlayacağı esnada bazı komutlar işletmek isteriz, bunu finalize metodunu kullanarak yaparız.□
- Finalize metodunun çalışması için sadece çöp toplama aracının devreye girmesi gerekiyor. Yani bizim bu metodu ayrıca çağırmamız gerekmiyor. Çöp toplama aracı devreye girince otomatik olarak bu metod çağrılacaktır ve bloktaki komutlar işleyecektir.

Aşağıda kullanımına ilişkin bir örnek verilmiştir.

```
public static void main( String[] args) {  
Personel personel = new Personel(); //nesne oluşturuldu  
new Personel();
```

```
//isimsiz bir nesne oluşturuldu

System.gc(); //çöp toplama aracı tetiklendi
}

public void finalize() { //finalize metodu
System.out.print("Isimsiz nesne hafızadan siliniyor");
}
```

Programda çöp toplama aracı çağrılıyor. Ancak çöp toplama aracı devreye girerse, sınıf içerisinde yazdığınız finalize metodu çalışacaktır. Özellikle System.gc komutu ile tetiklediğimiz halde, uygulamalarda finalize çıktısı oluşmayabilir. Biz çöp toplama aracını çağırmasak da, Personel sınıfına ait nesne silindiğinde finalize metodu çalışacaktır. Tüm anlatılan konuların içinde bulunduğu bir örnekle yazımızı sonlandırıyorum.

Örnek:

[crayon-5c694c077922a150628658/]

Örneğin Ekran Çıktıları:

```
Source History [Icons]
1 package personel;
2 public class Personel {
3
4
5     public static void main(String[] args) {
6         Personel personel = new Personel(); //nesne oluřturuldu
7
8         new Personel(); //isimsiz bir nesne oluřturuldu
9
10        System.gc(); //çöp toplama aracı tetiklendi
11
12    }
13    public void finalize() { //finalize metodu
14
15        System.out.print("Isimsiz nesne için hafızadan siliniyor");
16
17    }
18 }
19
```

```
personel.Personel >
Output - Personel (run) x
run:
Isimsiz nesne için hafızadan siliniyorBUILD SUCCESSFUL (total time: 0 seconds)
```

[SUNUM](#)

HAZIRLAYAN:

Beyza Nur Kılıç

Kaynakça: Kodlab – Java 8

```
[contact-form][contact-field
label='İsim'                type='name'
required='1']/][contact-field  label='E-
Posta'                type='email'
required='1']/][contact-field  label='Web
Sitesi'                type='url']/][contact-field
label='Yorum'                type='textarea'
required='1']/][/contact-form]
```

Difference Between Eclipse and Netbeans



James Gosling Java'yı geliştirirken temel olarak yarattığı bu dilin açık kodlu, nesneye yönelik, zeminden bağımsız, yüksek verimli, çok işlevli, yüksek seviye, adım adım işletilebilme gibi bir çok özelliği bir arada bulundurmasını amaçlıyordu. İlk başlarda bu amacına ulaşmasada zamanla Java onun hayal ettiğinden daha büyük kitlelere hitap edip, daha geniş bir kütüphaneye sahip oldu. Sun'ın 1995'te Java'yı duyurmasından kısa bir süre sonra yazılım geliştiriciler Java'yı küçük çaplı projelerde kullanmaya başladılar. Fakat özellikle

Java'nın bilgisayar mimarisine bağılı olmadan herhangi bir Java Virtual Machine(JVM)'de çalışabilme özelliğı bu dili son yüzyılın gözdesi haline getirdi.

Sun tarafından geliştirilen bir IDE olan NetBeans Eclipse kadar popüler olmasada Eclipse'in tahtını sallayacak kadar güçlü bir alt yapıya sahip. Profesyonel masaüstü, kurumsal, web ve mobil uygulamaları geliştirmek için gerekli tüm araçları içeriğinde barındırır. Windows, Linux, Mac OS X ve Solaris'te dahil olmak üzere birçok platformda çalışır. NetBeans'in en önemli özelliğı "All in One" sloganıyla yola çıkmış olmasıdır. Yani sizi gereksiz plugin indirip kurma zahmetinden kurtarır. Ayrıca profesyonel yazılımcıların yanı sıra amatör yazılımcılarda hitap eder. Özellikle masaüstü uygulamalarında designer'ı ile yazılım dünyasının sınırlarını zorlar. NetBeans'in bana göre en can alıcı noktası Visual Web Pack özelliğine sahip olmasıdır. Yazılımcı için sürükle bırak teknolojisini içeriğine katan NetBeans Java Derlemek için en ideal IDE olmasına rağmen bir çok eksiğinin olduğunu söylemem gerek. Bu eksikliklerden en önemlisi konfigürasyon dosyasını kendi kendine yaratması. Ayrıca yarattığı bu konfigürasyon dosyası üzerinde değışikliğe izin vermemesi yazılımcı için ilerki evrelerde büyük sıkıntılar yaratabiliyor. Gelelim Eclipse'e. IBM'in gözdesi olan Eclipse 2001 yılında duyurulduğunda Java dünyasında yer yerinden oynamıştı.(swing yerine swt'yi kullanması büyük tartışmalara neden olmuştu). Eclipse'in NetBeans'e göre en büyük avantajı popüler olmasıdır. Yani yaygın bir kullanım ağına sahiptir. Plugin bakımından inanılmaz zengin olmasına rağmen eğer ne yapacağınızı bilmiyorsanız plugin havuzunda boğulmanıza neden olur. Bu yönüyle amatör yazılımcılara değil profesyonel yazılımcılara hitap eder. Ayrıca Eclipse'te yazılımcı her adımı kendi tasarlamak zorundadır. Kendi konfigürasyon dosyasını yaratmalı kendi web.xml'ini oluşturmalıdır. Eclipse'in bu kadar geniş bir kitleye hitap etmesinin en büyük nedenlerinden biri de çok güçlü bir grafiksel kullanıcı arayüzü düzenleyicisine sahip olmasıdır.

Anahtar Farklılıklar

- Eclipse, zamandan tasarruf sağlayarak otomatik olarak Java kaynakları üzerinde biçimlendirme yapabilir. Bu özellik NetBeans 7.3'te yayınlandı.
- Eclipse java kod biçimlendiricisi Netbeans'dan çok daha güçlü görünüyor.
- NetBeans, farklı dizinlerde projeler açma kabiliyetine sahiptir.
- NetBeans, herhangi bir Maven projesini Eclipse'e özgü bir projeye dönüştürmek zorunda kalmadan açabilir.
- Netbeans maven entegrasyonu, maven dışındaki bir eseri (pom.xml olmadan) yerel (ve belki de diğer) maven depolarına kurma özelliğine sahiptir. Eclipse'de bu özellik yok. Sadece geçici çözümler var.
- Eclipse, VCS (ör. SVN, CVS vb.) ile yerel çalışma kopyası arasındaki farkları ağaç görünümü olarak gösteren bir "senkronize" görünümüne sahiptir. Netbeans da bir çeşit senkronizasyon görünümü vardır, ancak bilgileri bir tablo olarak görüntüler.
- NetBeans, Oracle tarafından desteklenmektedir. Eclipse, IBM sponsorluğundadır.
- NetBeans kullanıcı arabirimi Swing (Java yerli hafif araç seti) üzerine inşa edilmiştir.
- Eclipse kullanıcı arabirimi, SWT'ye (sistemin temel araç takımının etrafında bir Java sarmalayıcı) kurulmuştur; bu nedenle, platforma bağımlı derlenmiş ikili kitaplara ihtiyaç duyar.
- İkisi arasında platform desteği altında fark yok. Eclipse ve NetBeans, çapraz platform desteğine sahiptir. JVM (Java Virtual Machine) kurulu olduğu sürece, bu uygulamanın Windows, Mac, Linux, Solaris ve diğer platformlarda çalışmasını sağlayabilirsiniz.
- Her ikisi de C / C ++, Java, JavaScript ve PHP içeren geniş bir programlama dili desteği içerir. Ama nasıl bu desteği alırsınız ilginç bir parçasıdır. Eclipse, bir eklenti tabanlı IDE'dir. Onun işlevselliğinin büyük bir

kısmı eklentilerden geliyor. Mobil uygulama SDK'ları, Zengin İnternet uygulamaları ve Mimari odaklı uygulamalar gibi özellikler çoğunlukla eklentiler kullanılarak geliştirilebilir. Öte yandan NetBeans birçok projeye sahiptir ve araç tabanlı bir IDE'dir. Takımlama desteğini kullanarak birçok platformu içerir. Böylece daha az dağılmış hale geliyor.

Sonuç olarak eğer java programlama dilinde belli bir seviyenin üstündeyseniz ve alanda kariyer planları yapıp ün kazanma peşindeyseniz Eclipse IDE'sini kullanmanız sizin için daha iyi bir tercihtir. Fakat temelden başlayıp bu alanda kendinizi ilerletmek ve geliştirmek istiyorsanız Netbeans IDE'si sizin için daha iyi bir tercih olacaktır.

Yazar: Tayyip Doğan

Kaynaklar:

- <http://www.teknolojioku.com/haber/hangisi-dogru-ide-netbeans-mi-eclips-mi-876.html>
- <https://gelecegiyazanlar.turkcell.com.tr/soru/eclipse-mi-android-studio-mu>
- <http://www.differencebtw.com/difference-between-eclipse-and-netbeans/>
- <http://www.thewindowsclub.com/difference-eclipse-netbeans>

C++ EDİTÖRLERİ

C++ EDİTÖRLERİ

C++ dilinde program yazabilmek için çeşitli editörler vardır.

Bunlardan ücretsiz olanları örnek olarak biraz aşağıda göreceksiniz. Aslında C++ kodları yazmak için NOT DEFTERİ gibi çok basit bir editörde kullanılabilir. VISUAL STUDIO gibi çok gelişmiş ve karmaşık bir editör de kullanılabilir.

Ücretsiz derleyiciler

[Turbo Pascal ve Turbo C](#) :Borland ünlü Turbo Pascal'ın 1.0, 3.2 ve 5.5 sürümleri için serbest bıraktı.

[Dev-Pascal](#): Ücretsiz IDE ve Pascal için derleyici ([Free Pascal ile](#)).

[Dev-C ++](#) : C ve C ++ dilleri için ücretsiz IDE ve derleyici. Delphi ve C kaynak kodu mevcut.

[SmallC](#) : JE Hendrix [tarafından](#) yazılmış küçük C [derleyicisi](#) . C kaynak kodu dahil.

[BCX](#) :Lcc-Win32 ve MingW32 (veya Dev-C ++) altında herhangi bir değişiklik yapılmadan derlenebilir C kodu üreten çok güçlü ve kullanımı kolay bir sistem. Ayrıca, MS kaynak kodunu okuyabilen ve C'de (Win32) kod üreten bir kaynak çeviriciyle birlikte gelir.

[CoPascal](#) :Co-Pascal, N. Wirth tarafından geliştirilen Pascal-S derleyicisinin bir uzantısıdır. Pascal kaynak kodu dahil.

[P32](#) : Pascal derleyicisi. Şu anda geliştirme aşamasında ama harika çalışıyor. Pascal kaynağı dahil.

[PowerPascal](#) : Power Pascal, OS / 2 2.0 veya daha [yeni sürümü](#) için tamamen 32-bit, yerli Pascal [derleyicisidir](#) . Pascal kaynağı dahil.

[Djgpp](#) : C, c ++, ileri, pascal ve daha fazlasıyla C kaynakları içeren ücretsiz derleyici.

[TinyPascal](#) : Pascal dilinin küçük bir uygulaması. Tam Delphi

kaynakları dahildir.

[Ücretsiz Pascal](#): Dos, Linux, OS / 2 için 32-bit Pascal derleyicisi. Pascal kaynak kodu dahil. Dev-Pascal'da kullanılır.

[ScriptBasic](#) : Tam gömülebilir ve genişletilebilir kodlama temel çeviriciyi tam C kaynak koduyla. Windows NT ve Linux üzerinde test edilmiştir.

[LCC-Win32](#) :Windows için Jacob Navia'nın ücretsiz bir derleyici sistemi.

[Lcc](#) :lcc, ISO Standart C için yeniden hedeflenebilir bir derleyicidir. ALPHA, SPARC, MIPS R3000 ve Intel x86 ve ardılları için kod üretir.

[Yabasic](#): Windows ve Linux için küçük temel tercüman (Visual C ++ 6 için kaynak kodu ile)

[DOS için Pasifik C](#) : Profesyonel bir C IDE / derleyici shareware MSDOS için freeware sürümü

[TopLogo ++](#) : Geliştiriciler ve bilim insanları için bir IDE / derleyici aracı. Pakette tam belgeler, Derleyici IDE, demolar, yardım vb ... bulunur

[JavaTM 2 SDK](#) : JavaTM Yazılım Geliştirme Setleri ve Çalışma Zamanlıları

[XBasic](#) : etkileşimli program geliştirme ortamı, 32 bit / 64-bit TEMEL, etkileşimli grafik GuiDesigner, çok platformlu taşınabilir kaynak kodunu geliştirmiş, Windows95 – Windows98 – WindowsNT – Linux – UNIX

[XSCompiler](#) : Bu MSDOS derleyici, C ile uyumlu olan ancak sınıfları ve çoklu kalıtım içeren bir dilden 32 bit korumalı mod programları üretir. Standart C kütüphanesi, grafik, ses, çoklu görev, sıkıştırma, animasyon, GIF, PCX, FLI / FLC ve diğer kütüphaneler ile birlikte gelir. Derleyici bu dili

kullanarak derlenmiştir.

[GNAT](#) : GNAT, Ada95'de kaynak kodu bulunan bir Ada95 derleyicisidir.

[Rapid-Q](#) : Rapid-Q, Win32, Linux, Unix için ücretsiz bir IDE ve BASIC op-kodu derleyici sistemi ... Kullanımı çok kolay ve mySQL, cgi, DirectX, Direct3D, GTK gibi pek çok güzel ürünle bütünleşmek kolaydır ...

[Mingw](#) : GNU GCC [derleyicisinin](#) çok iyi bir Win32 portu (Dev-C ++'da [kullanılır](#))

[Cygwin](#) : GCC ve GNU Utils'in ücretsiz ve iyi bir Win32 portu

[GCC](#) : Tek ve tek GCC derleyici sistemi. Hemen hemen her sistemde çalışır.

[Phoenix / Envelop](#) : Basic altında hızlı gelişme.

[B ++](#) : B ++, C ++ üzerine kurulmuş bir BASIC derleyicidir – BASIC kodunu C ++ 'ya dönüştürür ve EXE (veya DLL) üretmek için C ++ derleyicisini çağırır. Kaynak kodu Free Pascal'da.

[GNU Ada](#) : Ücretsiz ADA derleyicisi.

[Dev86 / BCC](#) : x86 C / Assembler geliştirme araçları (C derleyicisi, assembler, linker). GPL altında. Linux ve DOS ikili dağıtımları mevcut.

[Http://www.digitalmars.com](http://www.digitalmars.com) : DOS, Win ve NT için Zortech C ++ kullanıcısının özgür C / C ++ Derleyicisi.

[Sanal Pascal](#) : Pascal dilini kullanarak 32-bit çapraz platform geliştirme için tercih edilen bir araçtır. Çalışma Zamanı Kitaplığı (RTL), optimize bir derleyici, güçlü bir entegre hata ayıklayıcı ve kapsamlı çevrimiçi belgeler de dahil olmak üzere Borland Pascal ve Delphi ile uyumludur.

[PlugSys Xbase derleyicisi](#) : PlugSys'in web sitesinde ücretsiz bir sürümü ve bir Professional sürümü vardır. Xbase (dbase,

clipper) dili için karakter tabanlı bir derleyicidir. DOSWin95 / 98 / NT ve Linux sürümü vardır. Aynı kod her iki sistemde de derlenir. Ayrıca bir sunucu sayfası aracı (ücretsiz ve ücretli sürümler) vardır. Karşıdan yükleme ile birlikte gelen belgeler harika. Ücretsiz sürüm yalnızca indirmek için kayıt gerektirir.

[Bağlam programlama dili](#) : Kaynak ve örnek programlarla MS-DOS için basit programlama dili ve derleyicisi, kendi giriş dilinde Andrei V. Khokhlov

[Babya](#) : Kurulum ve Windows 9x desteği ile DOS için B ++ derleyicisi.

[OpenWatcom](#) : Open Watcom, SciTech Software, Sybase® ve seçkin bir ekip ekibi tarafından Sybase Watcom C / C ++ ve Fortran derleyici ürünlerini Açık Kaynak topluluğuna taşıyacak ortak bir çabadır.

[Kendilik](#) : Kendini programlama dili için Linux'a yönelik bir derleyici portu. Sistem başlangıçta Stanford Üniversitesi ve Sun'da geliştirildi ve Java Hotspot derleyicisinin teknolojik temelini oluşturdu. Güzel bir GUI dahildir.

[SmallEiffel](#) : Eiffel dili için GNU [derleyicisi](#) . ANSI C derleyicisi gerektirir. AIX, Amiga, BeOS, BSD, Cygwin, MSDOS, FreeBSD, GNU / Hurd, GNU / Linux, HP-UX, IRIX, MacOS, NetBSD, NeXT, OS / 2, OSF1, QNX, SCO, Solaris, OpenVMS, Windows'u destekler 95/98 / NT / 2K ve XENIX.

[Inno Pascal](#) : Inno Pascal, Win32 için basit bir Pascal derleyicisidir. Herhangi bir harici assembler veya linker gerektirmeden kompakt, yerli kod x86 yürütülebilir dosyaları üretir. Tamamen sıfırdan yazıldı; Mevcut diğer derleyicilere dayanmaz. Tam kaynak kodu GPL lisansına dahildir.

[Liman Projesi](#) : Liman, Open Source Clipper Compatible Derleyicidir. OO uzantılarını ve birçok yararlı eklenti içerir. Lisans, GNU + Liman İstisnası'dır, bunun anlamı

lisansın derleyici ile ürettiğiniz programlara zarar vermediğidir.

Biz Dev-C++ kullanmayı tercih ediyoruz. Çünkü programlamaya yeni başladık , c++ da çok karmaşık olmadığından dolayı anlamamız daha kolay oluyor. Tavsiye ederim siz de yeni başlayacaksanız eğer...

YAZAR:Fatma AKTAŞ

Java ile OpenGL (JOGL) 'e Giriş

1. OpenGL nedir ?

OpenGL (Open Graphics Library) gelişmiş donanım desteğini kullanarak hem iki hemde üç boyutlu grafikleri ekrana çizdirmek için kullanılan ücretsiz bir grafik uygulama geliştirme arabirimidir. Windows, Linux, MacOS ve Solaris gibi birçok işletim sisteminde desteklenir. Donanım tarafında ise SGI, ATI, Nvidia ve Intel gibi büyük donanım üreticileri ekran kartlarında OpenGL desteği sunarlar.

1992 yılında ilk taslağı yaratılmış olan bu standart, günümüzde 4.5 sürümüne ulaşmıştır ve 700 üzerinde fonksiyona sahiptir. Çoklu platform desteği içeren uygulamalar ve özellikle de deneysel ve bilimsel araçlarda açık arayla önde ve standart olarak kullanılmaktadır.

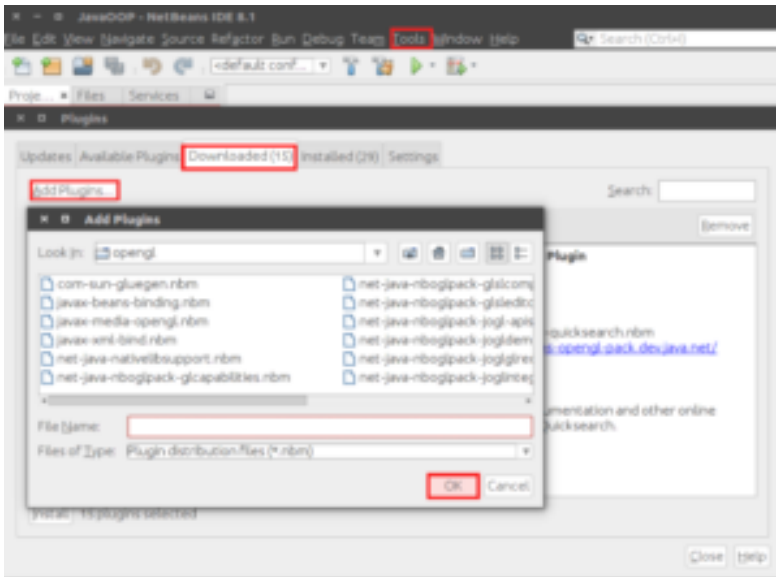
2. OpenGL eklentilerinin Netbeans'a kurulumu

OpenGL arabirimini netbeans'e eklemek için JOGL eklentisini indirmeliyiz.

JOGL'u şu adresten indirebilirsiniz :
<http://plugins.netbeans.org/plugin/3260/netbeans-opengl-pack>

İndirdiğiniz dosyayı zip'ten çıkarın. Netbeans'te Tools menüsü altında Plugins'ı seçin.

Açılan pencerede, Downloaded sekmesine gelip Add Plugins'e tıklayın. Zip'ten çıkardığımız NBM dosyalarının hepsini seçip Open'a basın.



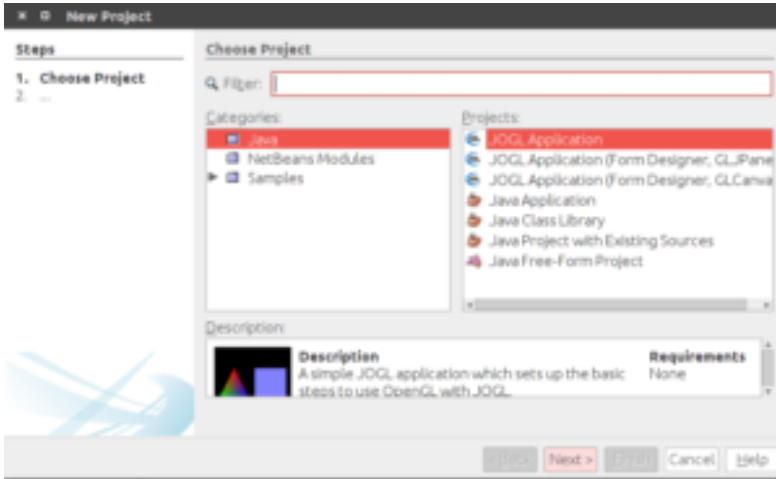
JOGL pluginlerinin netbeans'a eklenmesi

GLSL Editor dışındaki tüm eklentileri seçili hale getirip install'a basın. Sözleşme'yi kabul ettikten sonra Netbeans

tekrardan başlatılacak. Netbeans başladıktan sonra artık kullanıma hazırız.

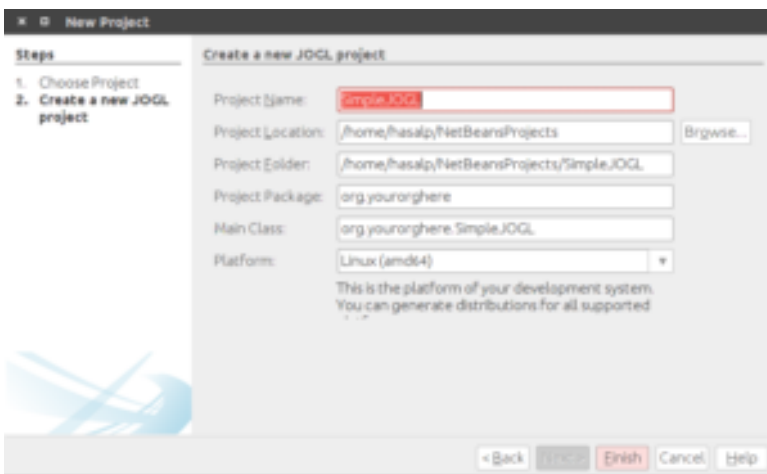
3. Netbeans JOGL eklentisinin kullanımı

Proje oluşturma ekranını açtığımızda JOGL Application'u seçiyoruz ve Next'e tıklıyoruz.

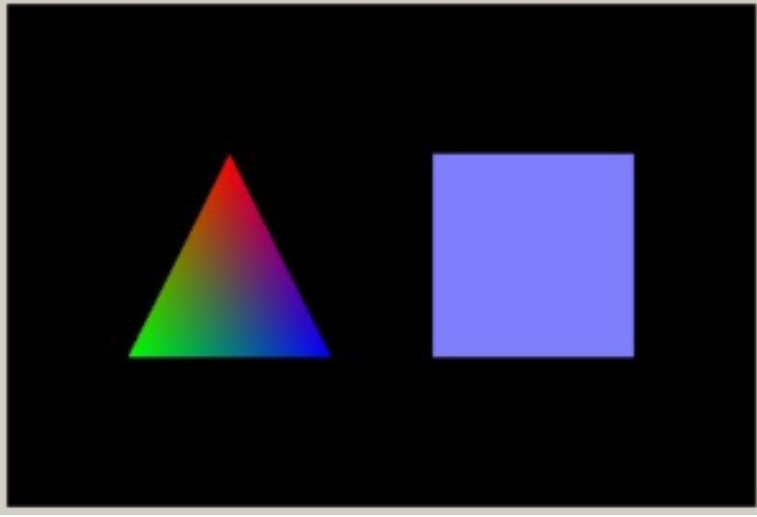


Projenin JOGL uygulaması olarak oluşturulması

Ardından projemizin ismini ve hangi platform'da çalışacağını seçiyoruz.



Projenin isminin ve platformunun seçilmesi.



Default programın çıktısı

Finish'e tıkladığımızda gerekli dosyaların import edilmiş olduğunu ve kod yapılarının oluşturulduğunu göreceğiz. Projemizi derlediğimizde default olan üçgen ve kare'yi göreceğiz.

Artık kendi kodlarımızı oluşturmaya hazırız.

4. Default Kodun İncelenmesi

[crayon-5c694c077a0e1929390332/]

Yazar: Hasan Alp ZENGİN

Kaynaklar:

- www.ahmetsayar.com/wp-content/uploads/Java-ile-OpenGL-Giriş.pdf
- www.en.wikipedia.org/wiki/Java_OpenGL