

FURKAN

ÖZEV

161044036

1. Derleyici (compiler) ile çevirici (assembler) arasındaki fark nedir?

Compiler, bir dilde yazılmış bir programı okuyan ve onu başka bir dile çeviren bir bilgisayar programıdır, ancak bir Assembler yalnızca Assembly dilini makine koduna çeviren özel bir derleyici türü olarak kabul edilebilir.

Derleyiciler (Compilers) genellikle makinenin çalıştırılabilir kodunu doğrudan üst düzey bir dilden üretirler, ancak çeviriciler (assemblers), bir makinede çalıştırmak için bağlayıcı programları (linker programs) kullanılarak bağlanması gerekebilecek bir nesne kodu üretir.

Assembly dili, makine koduna bire bir çeviri yaptığından, bir assembler, performansın çok önemli olduğu durumlarda kullanılır.

2. Aşağıdaki soruları cevaplayınız.

a) Fonksiyonel (functional), emirsel (imperative) ve bildirimsel (declarative) programlama dillerini kısaca açıklayınız.

Fonksiyonel (functional): Bu paradigma, bir programı girdileri kabul eden ve çıktılar üreten bir varlık olarak görülüyor. Bu paradigma ile önceden tanımlı daha küçük program birimlerini (önceden tanımlı işlevleri) bağlayarak bir program oluşturulur, böylece her birimin çıktıları istenilen toplam girdi-çıkı ilişkisi elde edilecek şekilde başka birimin girişleri olarak kullanılır. Kısacası, işlevsel paradigma altındaki programlama süreci, basit işlevlerin iç içe geçmiş kompleksleri gibi yapı fonksiyonlarıdır. Bu paradigma kod tekrarını engellemekte ve aynı kodun farklı şartlar için tekrar tekrar çalışmasını sağlamaktadır aynı zamanda kodun okunabilirliğini arttırmakta ve kod analizini daha kolay hale getirmektedir.

Emirsel (imperative): Emirli programlama ile kastedilen, bir programlama dilindeki komutların satır satır emirlerden oluşmasıdır. Emirli programlamada nasıl sorusuna cevap olacak komutlar ön plana çıkar. Yani yaptırmak istediğimiz işi nasıl yapılacağını emir komutlarıyla yaptırırız. Emirli programlamada hedef emirlerin doğru şekilde ve dizilimde verilmesidir. Emirli programlamadaki her fonksiyon belirli bir başka emirin alt parçalarıdır ve sürekli olarak programcının emir vermesi gerekir. Emirli programlama sanki robotun her işleminin emre bağlanması olarak düşünülebilir.

Bildirimsel (declarative): Bildirge niteliğindeki programlama, zorunlu olmayan herhangi bir programlama tarzı olarak tanımlanır. Bu programlama yaklaşımında yazılan her fonksiyon birer amaca hizmet eder ve daha sonra bu fonksiyonlar birleştirilerek modüler bir problem çözümü elde edilir. Yani tanımlamalı programlamada ne sorusuna cevap olacak komutlar ön plana çıkar. Bu anlamda tanımlı programlama sanki robotumuzun daha çok şey bilmesini sağlar. Bildirimsel paradigma matematiksel mantığa açıkça yazılabilen bir dildir. Bir hesaplamanın ne yapması gerektiğini açıklayan üst düzey bir program bu paradigmaya örnektir.

b) C, SQL, C#, F#, Haskell, Prolog, Common Lisp, Java dilleri yukarıda geçen dillerden hangisine girer?

Fonksiyonel (functional)	Emirsel (imperative)	Bildirimsel (declarative)	Object-oriented (nesne yönetimli)
F#	C	SQL	C#
Haskell	Java	Prolog	Java
			Common Lisp

c) Common Lisp ve SQL dillerinin tarihçeleri, avantajları, dezavantajları ve nerelerde kullanıldıkları hakkında kısaca bilgi veriniz

Common Lisp: Common Lisp, Lisp programlama dilinin bir lehçesidir. Common Lisp, MacLisp lehçelerinin özelliklerini birleştirmeye, standartize etmeye ve genişletmeye çalışmak için 1980 li yıllarda ANSI X3J13 komitesi tarafından geliştirilmeye başlandı. Common Lisp uygulamadan ziyade bir dil belirtimi olarak ifade edilir. Yani genel amaçlı, çok paradigma programlama dilidir. Prosedürel, işlevsel ve nesneye yönelik programlama paradigmlarının birleşimini destekler. ARPA yöneticisi Bob Engelmores tarafından bir Lisp lehçesi geliştirmek için 1981’de Common Lisp çalışması başlatıldı. İlk dil tasarımının çoğu elektronik posta ile yapıldı. Common Lisp in ilk genel görünümü 1982’de verildi. İlk Dil Dökümantasyonu (Common Lisp The Language) baskı olarak 1984’te basıldı. 1990’da basılan 2.baskı ile birçok değişiklik dile eklendi. Son olarak 1994’te ANSI Common Lisp standardı yayınlandı. Uygulamalar ve kütüphaneler tarafından Common Lisp için çeşitli uzantılar ve geliştirmeler sağlanmıştır.

Common Lisp Avantajları - Dezavantajları:

Common Lisp dilini öğrenmek, C veya Java gibi dilleri öğrenmekten zor değildir. Basit ve esnek yazıma sahiptir. Değişkenleri liste olarak kabul edebilir ve programın çalışabilmesi için kendisi tanımlar. Girdileri her yönüyle değiştirebilir. Beklenmedik bellek hatalarını önler. Tek veri yapısına sahip bir dil olduğundan liste verilerini işleyemez. Eski bir dil olduğundan yavaştır. Daha çok yapay zeka veya akademik araştırmalar için kullanılır.

SQL: SQL’in ana kullanım yeri veritabanlarıdır. Verileri yönetmek ve tasarlamak için kullanılan bir veritabanı yönetim sistemidir. SQL, kendisi bir programlama dili olmamasına rağmen birçok kişi tarafından programlama dili olarak bilinir. İlk başta matematiksel bir sözdizimine sahip olan SQUARE adlı bir dil geliştirilmiştir. Geniş kullanıcı kitleleri tarafında kolay kullanılabilmesi için matematiksel söz dizimli SQUARE dilinden vazgeçilerek, İngilizce’ye benzer sözdizimine sahip bir dil oluşturulmuş ve SEQUEL olarak adlandırılmıştır. SQL İle sınırsız sayıdaki veri arasından istediğiniz veriyi bulabilmek, Sınırsız sayıdaki veri arasında istediğiniz her sorguyu yapabilmek, Veritabanına yeni veri kaydetmek, Veritabanında var olan verileri çekmek, düzenlemek, güncellemek ve silmek, Veritabanı içinde yeni tablo oluşturmak, Yeni veritabanı eklemek ya da var olanı değiştirmek gibi işlemleri yapabilirsiniz.

SQL Avantajları:

1-Hız: Bir veritabanından büyük miktarda kayıtları hızlı ve verimli bir şekilde alınabilir.

2-Standartlar: ANSI & ISO tarafından benimsenen uzun zamandır kurulmuş standartları kullanmaktadır.

3-Kodlama Yok: Standart SQL kullanarak, önemli miktarlarda kod yazmak zorunda kalmadan veritabanı sistemlerini yönetmek daha kolaydır.

SQL Dezavantajları:

1-Arabirimde Zorluk: Bir SQL veritabanına bağlanmak, birkaç satırlık kod eklemekten daha karmaşıktır.

2-Veri Tasıma: Bir veriyi başka bir ortama taşıırken bir çok problem ortaya çıkar.

3. Verilen değerler için işlemlerin sonucu nedir?

a) $4 * 6 + 24 = 48$

b) $27 / (6 - 3 * 1) = 9$

c) $4 * 6 / 2 + 17 \% 2 = 13$

d) $6 > 2 + 3 * 2 \&\& 0 = 0$

4. Aşağıdaki emirsel dil kodunu fonksiyonları verilen fonksiyonel dil koduna dönüştürünüz. Fonksiyonel dile ait fonksiyonların hepsini kullanmak zorunda değilsiniz. Dilerseniz bir fonksiyonu birden çok defa kullanabilirsiniz.

Emirsel Dil Kodu:	Fonksiyonel Dil Fonksiyonları:
vize_toplam = vizelerin toplamasi vize_ortalatasi = vize_toplam / 2 vize_agirlikli_notu = vize_ortalatasi * 0.6 final_agirlikli_notu = final_notu * 0.4 ders_notu = vize_agirlikli_notu + final_agirlikli_notu	(liste_topla girdi_listesi) (topla sayi1 sayi2) (cikar sayi1 sayi2) (carp sayi1 sayi2) (bol bolunen bolen) (Not: Kalan önemsizdir.)

CEVAP:

ders_notu = (topla(carp (bol (topla sayi1 sayi2) (liste_topla girdi_listesi)) 0.6) (carp final_notu 0.4))

5. Aşağıdaki fonksiyonel dil kodunu emirsel dil koduna dönüştürünüz. Burada faturaların elektrik, doğalgaz, su, telefon ve internet olduğunu kabul ediniz. Cevabınızı buna göre yazınız.

Fonksiyonel Dil Kodu: (cikar (topla eski_bakiye maas) (liste_topla faturalar))

CEVAP:

guncel_bakiye = eski_bakiye + maas

faturalar = elektrik + dogalgaz + su + telefon + internet

kalan = guncel_bakiye - faturalar