ENOCA SINAV JAVA

|  |  |
| --- | --- |
| Adınız: | ÜLKÜ |
| Soyadınız: | KÜLEKÇİ |
| Üniversite: | AKDENİZ ÜNİVERSİTESİ |
| Fakülte: | MÜHENDİSLİK FAKÜLTESİ |
| Bölüm: | ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ |
| Sınıf: | 1 / 2 / 3 / 4 / Mezun / Yüksek Lisans |
| Telefon: | 05346599190 |

**1) Back-end konusunda kendinizi hangi aşamada görüyorsunuz?**

Başlangıç / Orta / İyi / Çok İyi

**2) Front-end konusunda kendinizi hangi aşamada görüyorsunuz?**

Başlangıç / Orta / İyi / Çok İyi

**3) Veritabanı konusunda kendinizi hangi aşamada görüyorsunuz?**

Başlangıç / Orta / İyi / Çok İyi

**4) En çok kullandığınız yazılım dillerini yazınız.**

**Java css html java spring java script**

**5) En çok kullandığınız veritabanı sistemlerini yazınız.**

Postgresql, mysql

**6) Bireysel veya Kurumsal bir projede bulundunuz mu? Detaylarını açıklayın.**

İnnova bilişim in java bootcamp

Turkcell java bootcamp

Engin Demiroğ java Kampına devam etmekteyim.

**7) MVC kavramını açıklar mısınız ? Neden ihtiyaç duyuluyor?**

MVC: Model-View-Controller Kullanıcı arayüzünü birbirinden ayırarak farklı amaçlar için kullanılan bölümleri ayrı ayrı gruplandırarak karmaşıklığı engellemeyi sağlar. Böylece sorumluluklar ayrılmış olur. Kısacası projemizi katmanlıyoruz. Veri tabanı için bir katman, sorgularımız için ayrı bir katman ve son kullanıcıya sunulacak olan ekran için ise ayrı bir katman oluşturmaktır. Model genellikle veri tabanı işlemlerimizi yaptığımız yapıdır. Veri tabanımız üzerinde yapılabilecek sorgularımızı burada belirler ve Controller e atamamızı sağlarız. Bu sayede veri tabanımıza dışarıdan daha kolay erişebilir ve çeşitli metotlarla daha kolay idare edilebilir hale getirir. Controller Model yardımıyla oluşturduğumuz sorguların kullanıcı tarafından alınan veriler ile birleşip uyumlu bir biçimde çalışmasını sağlayan yapıdır. View sonucun gösterildiği bölümdür.

**8) Bir web sayfasında ekran sürekli Backend’den veya bir başka yapı tarafından güncelleniyor. Siz, web sayfasını refresh etmeden bu güncel bilgiyi anlık olarak ekrana hangi teknoloji ile yansıtırsınız.**

**Java için:**

<dependency>

<groupId>com.azure.spring</groupId>

<artifactId>azure-spring-cloud-appconfiguration-config-web</artifactId>

<version>2.6.0</version>

</dependency>

<!-- Adds the Ability to Push Refresh -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

Böyle bir dependency ekleyerek.

**9) String tipinde sabit bir şifre değişkeni oluşturun. Örneğin: String password = “12345”; gibi. Ardından, kullanıcıdan bir şifre girmesini isteyin. Girilen şifre ile değişkende tuttuğunuz değeri kıyaslayın. Eğer, iki değer birbirine eşitse ekrana “Giriş Başarılı!”, değilse “Giriş Başarısız” yazdırın.**

**(JAVA yazılım dilinde yazınız)**

|  |
| --- |
|  |

**10) Aşağıdaki görüntüyü ekrana yazdıran algoritmayı tasarlayınız**

**(JAVA yazılım dilinde yazınız)**

|  |  |
| --- | --- |
|  |  |

**11) Bir sayının Mükemmel bir sayı olup olmadığını bulan metod yazınız.**

**(JAVA yazılım dilinde yazınız)**

**Mükemmel sayı : Mükemmel sayı,sayılar teorisinde, kendisi hariç pozitif tam bölenlerinin toplamı kendisine eşit olan sayı. Diğer bir ifadeyle, bir mükemmel sayı, bütün pozitif tam bölenlerinin toplamının yarısına eşittir. 6=> 1 2 3 Mükemmel sayıdır.**

|  |
| --- |
|  |

**12) Linux üzerinde** **deneme.txt dosyasını herkese yazma hakkı nasıl verilir?**

chmod o+r deneme.txt

**13) İstediğiniz bir veritabanı sisteminde “Firma” ve “Personel” tablosu olsun. Aşağıda istenen sorguları yazınız.**

**Firma:** firmaID, firmaAd, firmaAdres, firmaWebSite

**Personel:** personelID, firmaID, personelAd, personelSoyad, personelPuan

-Firma Adı “Enoca” olan personelleri getirin

-Personel Adı “Ahmet” olan Personellerin personelPuanlarını 0 olarak güncelleyin.

-Personel Adı “Ahmet” olan Personelleri siliniz.

-Personel tablosuna bir personel ekleyin, Firması boş olsun

--PostgreSQL database dump

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SET check\_function\_bodies = false;

SET client\_min\_messages = warning;

SET default\_tablespace = '';

SET default\_with\_oids = false;

CREATE TABLE Firma (

firmaID smallint NOT NULL,

firmaAd character varying(20) NOT NULL,ABSOLUTE

firmaAdres character varying(15),

firmaWebSite character varying(15),

);

CREATE TABLE Personel (

personelID smallint NOT NULL,

firmaID smallint NOT NULL,

personelAd character varying(20) NOT NULL,

personelSoyad character varying(10) NOT NULL,

personelPuan character varying(10) NOT NULL,

);

INSERT INTO Firma VALUES (1, 'Enoca', 'Malatya', 'www1', '\x');

INSERT INTO Firma VALUES (2, 'Abc', 'Erzurum', 'www2', '\x');

INSERT INTO Firma VALUES (3, 'Def', 'Kars', 'www3', '\x');

INSERT INTO Firma VALUES (4, 'Sef', 'Antalya', 'www4', '\x');

INSERT INTO Personel VALUES (1, 1, 'Ahmet', 'Ak', '0', '\x');

INSERT INTO Personel VALUES (2, 2, 'Ayşe', 'Demir', '12', '\x');

INSERT INTO Personel VALUES (3, 3, 'Ali', 'Boz', 4, '\x');

INSERT INTO Personel VALUES (4, 4, 'Veli', 'Yapıcı', '5', '\x');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Select \* from Personel where firmaAd = 'Enoca';

UPDATE Personel SET personelPuan = '0', WHERE personelAd 'Ahmet';

DELETE FROM Personel WHERE personelAd = 'Ahmet';

INSERT INTO Personel VALUES (5, NULL, 'Engin', 'Külekçi', '0', '\x');

**14) Spring framework nedir? Birkaç cümle ile açıklayınız. Hiç Kullandınız mı?**

Kullandım bu anlamda githup da eklediğim örnekler var.

Java ile geliştirme yapmayı kolaylaştıran Core Container, AOP, Data Access, Web gibi modüllerden oluşan bir framework-kütüphanedir. Java tabanlı enterprise uygulamalar için kapsamlı bir programlama ve konfigürasyon altyapı desteği sunar. Spring Framework MVC mimarisini sunması, Aspect Oriented Programlama, Restful web servisleri, JDBC, JPA desteği gibi birçok özellik sağlamaktadır.

**15) Aşağıdaki kavramları açıklayınız**

**Pointer:** İşaretçi veya göstericiler değer olarak bellek adresini tutan değişkenlerdir.

Örneğin İnt \*ptr şeklinde tanımlama yapalım ve birde int a diye değişken oluşturalım sonra a değişkenine ptr yi atayalım böylece a nın yerini ptr hafızada tutacaktır. Böylece belleği yönetebiliriz. Silme güncelleme… gibi gibi

**Array:** Array(dizi), aynı veri tipinden çok sayıda değişken demektir. Çok sayıda verinin hızlı bir şekilde kullanılmasını sağlar. Tek boyutlu çok ve iki boyutlu diziler mevcuttur. int a1[] = new int[] = {20,21,22,23}; örnek bir array kullanımı array kullanmasaydık burada tek tek int a1, int a2, int a3 şeklinde değişkenler oluşturacaktık. Bu da bizim için hiç kullanışlı değil mümkün olduğunca kısa olmalı kodlarımız.

**TreeView:** Ağaç görünümü veya anahat görünümü , bilgilerin hiyerarşik bir görünümünü sunan grafiksel bir kontrol öğesidir . Her öğe (genellikle bir dal veya düğüm olarak adlandırılır ) bir dizi alt öğeye sahip olabilir. Bu genellikle bir listede girinti ile görselleştirilir.Bir öğe , varsa alt öğeleri ortaya çıkarmak için genişletilebilir ve alt öğeleri gizlemek için daraltılabilir .kullanıcının dosya sistemi dizinlerinde gezinmesine izin verdiği dosya yöneticisi uygulamalarında görülür . Ayrıca, bir XML belgesi gibi hiyerarşik verileri sunmak için de kullanılırlar .

**Soyutlama (Abstraction):**

Soyutlama kavramı sınıfın içindeki iç işleyişi dışarıdan izole etmek, yani gizlemektir. Örneğin: bilgisayarı kullanırken çoğu kullanıcı bilgisayarın iç işleyişinden haberi olmaz. Hafızanın işlemciyle haberleşmesi, işlemler arası senkronizasyon, klavyeden girilen değerlerin ekrana yansıması gibi birçok işlemin detayı kullanıcılardan gizlenmiş durumdadır. Kullanıcılar sadece bu fonksiyonları veya işlevleri bir arayüz vasıtasıyla çağırıp kullanmaktadır. İç detaylarına müdahale etmemektedir.

**Kapsülleme (Encapsulation):**

Sarmalama ilkesi, bir sınıfa ait değişkenlerin veya niteliklerin ancak o sınıfa ait metotlar tarafından değiştirilebilmesi ve okunabilmesi ilkesidir. Bu ilke sayesinde nesnelerde oluşacak anlamsızlıkların önüne geçilebilir. Ayrıca değişkenlere sınıfların dışından erişim olmaması ve bir sınıf içindeki değişkenlerin nasıl ve ne kadar olacağının da başka kodlardan saklanmış olması anlamına gelir. Böylelikle biz değişkenlerimizi sarmalayarak istenmeyen durumlardan korunacak bir filtre haline dönüştürebiliriz.

**Miras Alma (Inheritance):**

Kalıtım, programlama ortamında da gerçek hayattaki tanımına benzer bir işi gerçekleştirir. Bir sınıfın başka bir sınıftan kalıtım yapması demek, kalıtımı yapan sınıfın diğer sınıftaki nitelik ve davranışlarını kendisine alması demektir. Kalıtımı yapan sınıfa alt sınıf, kendisinden kalıtım yapılan sınıfa ata sınıf dersek, ata sınıfta tanımlı olan herşeyin alt sınıf için de tanımlı olduğunu söyleyebiliriz**.**

**Çok Biçimlilik (Polymorphism):**

Çok biçimlilik aynı görevin veya işin farklı yollarla yapılabilmesini ifade eder. Nesne, aynı davranışı farklı formlar ve görünüşler ile yerine getirebilir. Bunu yapabilmek için iki yöntem vardır.

Overriding in Java (Ezme)

Overloading in Java (Aşırı yükleme)