

Dikkat Edilmesi Gereken Konular

Burak Bayramlı

Bu belge, \LaTeX ile üretilmiştir

Konular

- Teknoloji Seçmek

Teknoloji Seçmek

- Kurumsal Uygulamalarda C++, CORBA vardı
- Şimdi Java EE
- Bu değişimin sebebi nedir?

Üç faktör

Üç faktör

- Basitlik
- Basitlik
- Basitlik

Detaylar

- Kurumsal programcılar bellek idaresiyle uğraşmak istemediler
- C++ dili en basit işi basit yapmanıza yardım etmiyordu
- .cpp dosyaları ve .h dosyaları birbirinden ayrı (niye tek class tek dosya olmasın?)

İlerisi İçin

- Çok kısa süre içinde teknoloji XYZ ile en basit örnek yazılabilmeli
- **Dili** temiz olmalı (IDE'sinin olup olmaması bonus, ama ana faktör değil)
- Daha kısa daha temiz dil, uzun vadede kazanır

Niye Dil?

- Yazılı/sözlü dil en sıkıştırılmış (compressed) iletişim yöntemi
- Dil, kelimelerin yanyana gelmesi (görüntülerin değil)
- Televizyon testi
 - ★ Sesi kapatın, görüntüden ne olduğunu anlamaya çalışın
 - ★ Sonra görüntüyü karartın, sestene ne olduğunu anlamaya çalışın

Eski Özdeyiş

“Bir resim bin kelimeye bedeldir”

Yeni Özdeyiş

“Kurumsal Uygulamalarda kod/simge/kelime, bin resme bedeldir”

Kod

- Form'da girilen bilgilerin ışığında bir obje yarat, ve başka bir objeye ver. Form'daki alanların hepsi nesne üzerindeki alanlara uymalı ve veri transferi ona göre yapılmalı

Kod

- Form'da girilen bilgilerin ışığında bir obje yarat, ve başka bir objeye ver. Form'daki alanların hepsi nesne üzerindeki alanlara uymalı ve veri transferi ona göre yapılmalı
- @In

Kod

- Bu class veri tabanına/tabandan yazılıp okunur hâle getirilmeli, içindeki her özellik eğer tersi belirtilmemişse verilen tablo içinde bir alana eşlenmeli.

Kod

- Bu class veri tabanına/tabandan yazılıp okunur hâle getirilmeli, içindeki her özellik eğer tersi belirtilmemişse verilen tablo içinde bir alana eşlenmeli.
- `@Entity`

Katmadeğer

- Bir teknoloji bir şeyler eklemeli
- *Toparlayan* teknolojiler uzun vadede kaybederler
- Meselâ Spring Framework
 - ★ Birçok J2EE teknolojisini “ayarlamamızı” sağlıyordu.
 - ★ Fakat: Katmadeğer nerede?
 - ★ Eğer daha basit ayarlanan teknolojiler kullanırsak Spring’e gerek var mı?

Kullanım Açısından Teknoloji Grupları

- Kontrolcü teknolojiler
- Köle teknolojiler

Köle Teknolojiler

- Kara kutu
- Sadece çağırılır
- O tek iletişim noktasından sonra sizin kodunuzda fazla etkisi olmaz
- Meselâ Java Mail

Kontrolcü teknolojiler

- Kodunuzun *tamamının* işleyişinde etkilidirler
- Kodunuz tarafından hem çağırılır, bazen onu *çağırırlar*, ve bazen onu değiştirirler

Seçim

- Uzun vadede kod bakımı açısından kontrolcü teknolojiler daha önemli
- Kodlamanın her aşamasını etkilediği için teknik lider iyi seçim yapmalı
- Spring, Java EE (annotation ayarları ile) kontrolcü teknolojidir

- Bunlardan bir tanesi katmadeđer sađlıyor, diđerini sađlamıyor

Ek Bilgiler

- Site: bilgidata.com adresinden en son kodları alabilirsiniz.
- Bu eğitimin kodları, 2. baskı kitap kodlarını baz aldı
- bilgidata.com/kurumsaljava sayfası alt bölümde bağlantılar var
- Bana erişmek için egitim@bilgidata.com adresini kullanabilirsiniz.

Son