



# JAVA 1

Enum, Design patterns,  
Generics

# Teme

- Enum
- Design patterns
  - Repository
  - Factory method
  - Singleton
- Generics

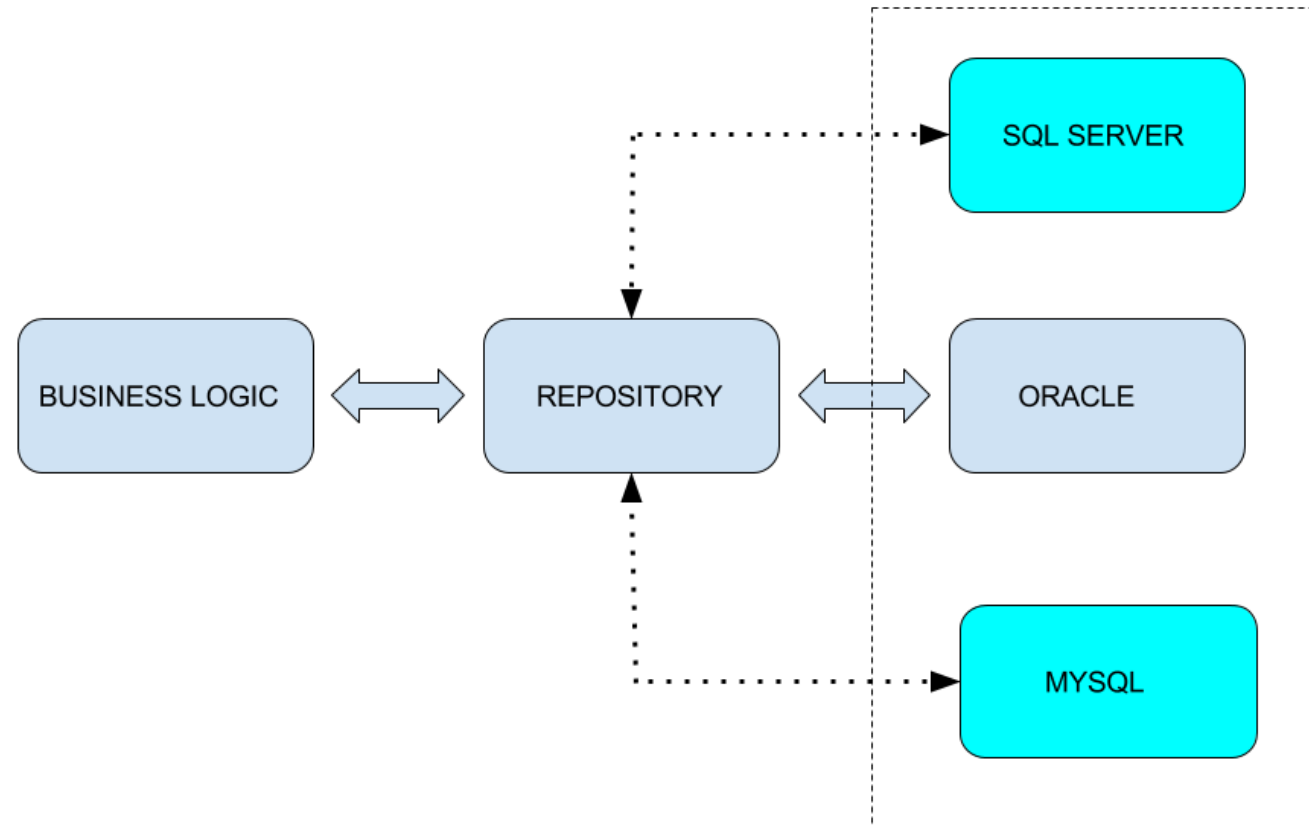
# Enum

- referentni tip
- sadrži pobrojani niz konstanti
- pogodan za *switch-case*
- konstante su implicitno *final* i *static*
- kreiranje instanci jedino tijekom deklaracije
- može sadržavati konstruktore, varijable...
- *values()* – dohvaća sve vrijednosti enumeracije
- implicitno najbolji Singleton

# Design patterns

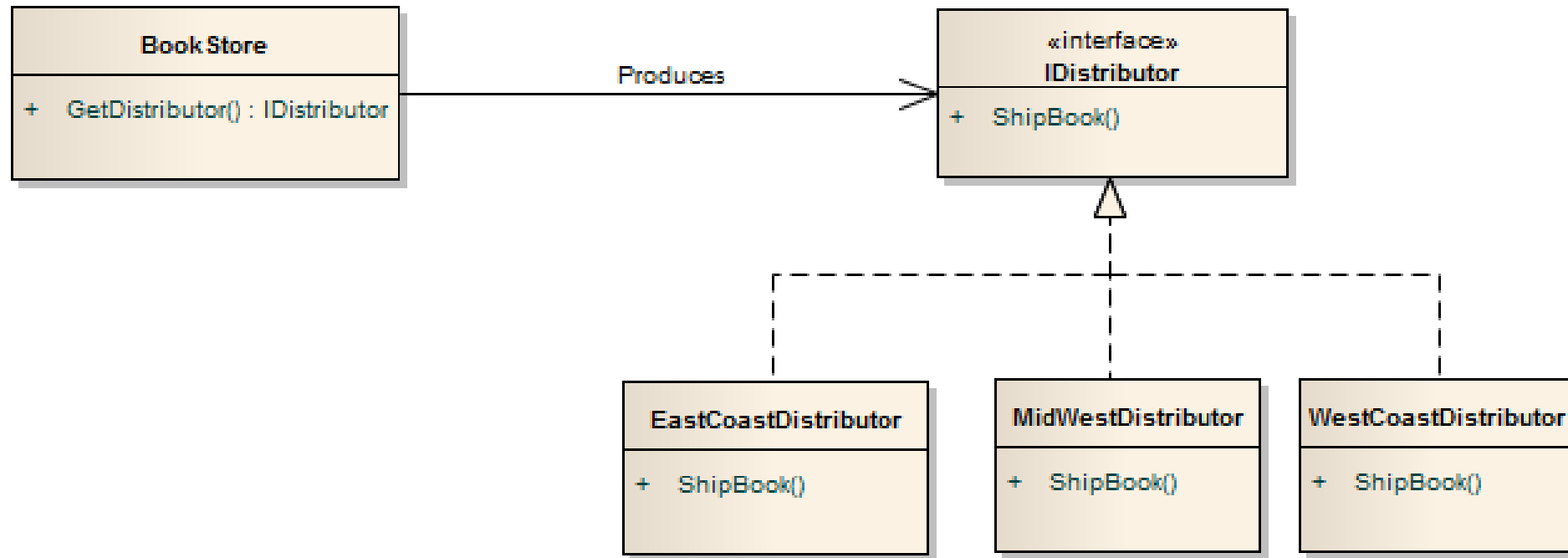
- općenita rješenja ponavljajućeg problema
- obrasci, nacrti rješenja, ne njihova implementacija
- testirane, dokazane paradigme pristupa
- ubrzavaju razvoj aplikacija
  - povećavaju čitljivost koda
  - smanjuju potrebu za promjenama
- ne smiju biti sebi svrhom!
- Repository pattern in C#
- Factory method pattern

# Repository Pattern



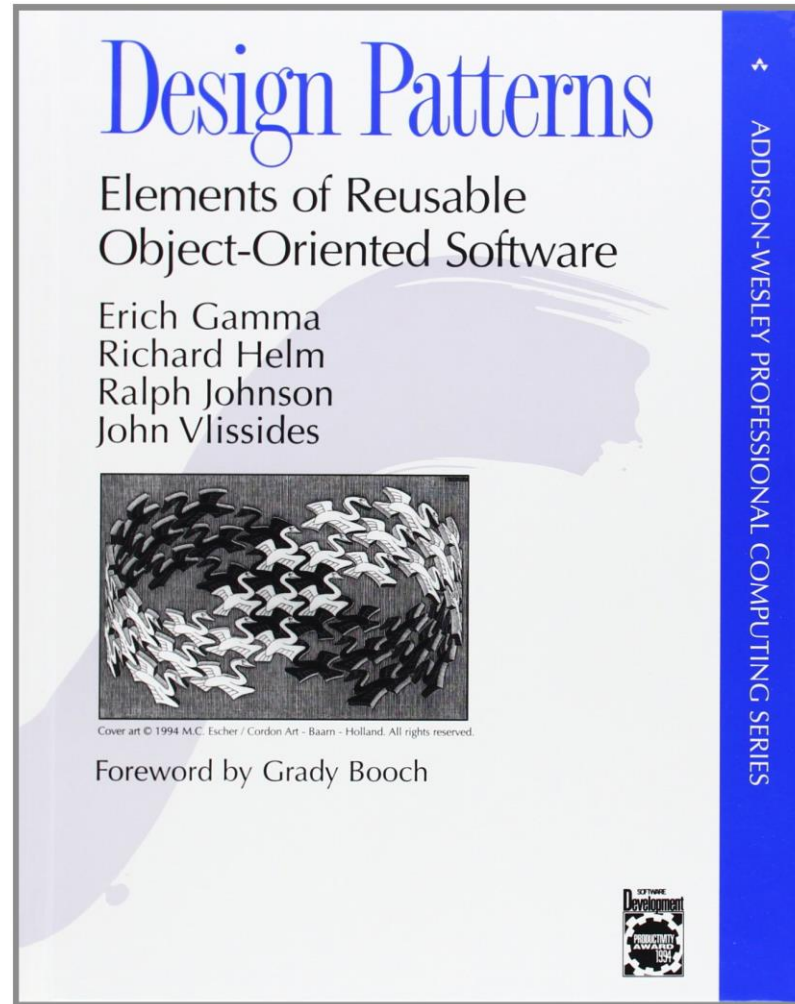
Izvor: <https://www.codecompiled.com/csharp/repository-pattern-in-csharp/>

# Factory method pattern



Izvor:<http://dotnetmother.blogspot.com/2014/04/factory-design-pattern.html>

# Design patterns

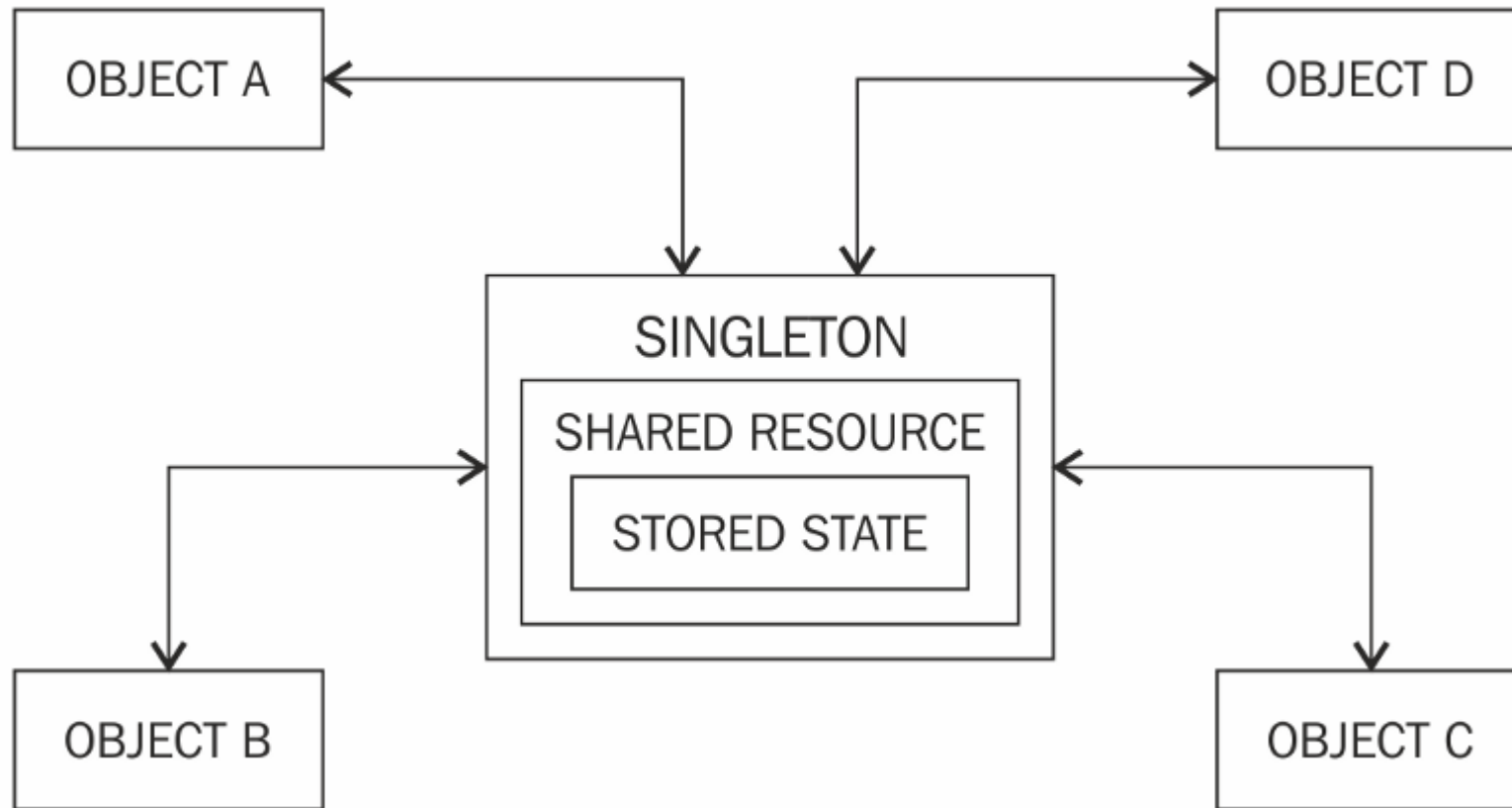


# Singleton pattern

- klasa za koju želimo samo jednu instancu
- privatni konstruktor
- statička referenca koja se kreira *lazy* ili *eager*
- *dobra strana*:
  - upravljanje stanjem pojednostavljeno – *configuration, logging*
- *loše strane*:
  - globalne instance
  - *tight coupling* – trebali bi ovisiti o apstrakciji ne o implementaciji
  - teže pisanje unit-testova



# Singleton pattern



Izvor: <https://www.freecodecamp.org/news/singleton-design-pattern-pros-and-cons-e10f98e23d63/>

# Generics

- poput *template* u C++
- parametriziranje metoda (parametri i povratne vrijednosti), klasa, sučelja
- izbjegavanje redundancije
- generiraju *compile-time* umjesto *run-time* greške
- moguće koristiti više tipova (T, U, R)
- moguće ograničavanje tipova (*extends*, *super*)

# Demo

- Project



<http://www.jnhsolutions.com/contact-us/request-a-demo/>

**Hvala na pažnji!**

