

# Objektni software design u ABAP-u

# Sadržaj

- Objektno programiranje i objekti
- Prednosti objektnog programiranja
- ABAP kao objektni jezik
- Podizanje kompetencija
- Uzroci potencijalnog neuspjeha
- Uzorci dizajna (Design Patterns)
- Stvarni primjer primjene
- Zaključak

# Sadržaj

- • **Objektno programiranje i objekti**
- Prednosti objektnog programiranja
- ABAP kao objektni jezik
- Podizanje kompetencija
- Uzroci potencijalnog neuspjeha
- Uzorci dizajna (Design Patterns)
- Stvarni primjer primjene
- Zaključak

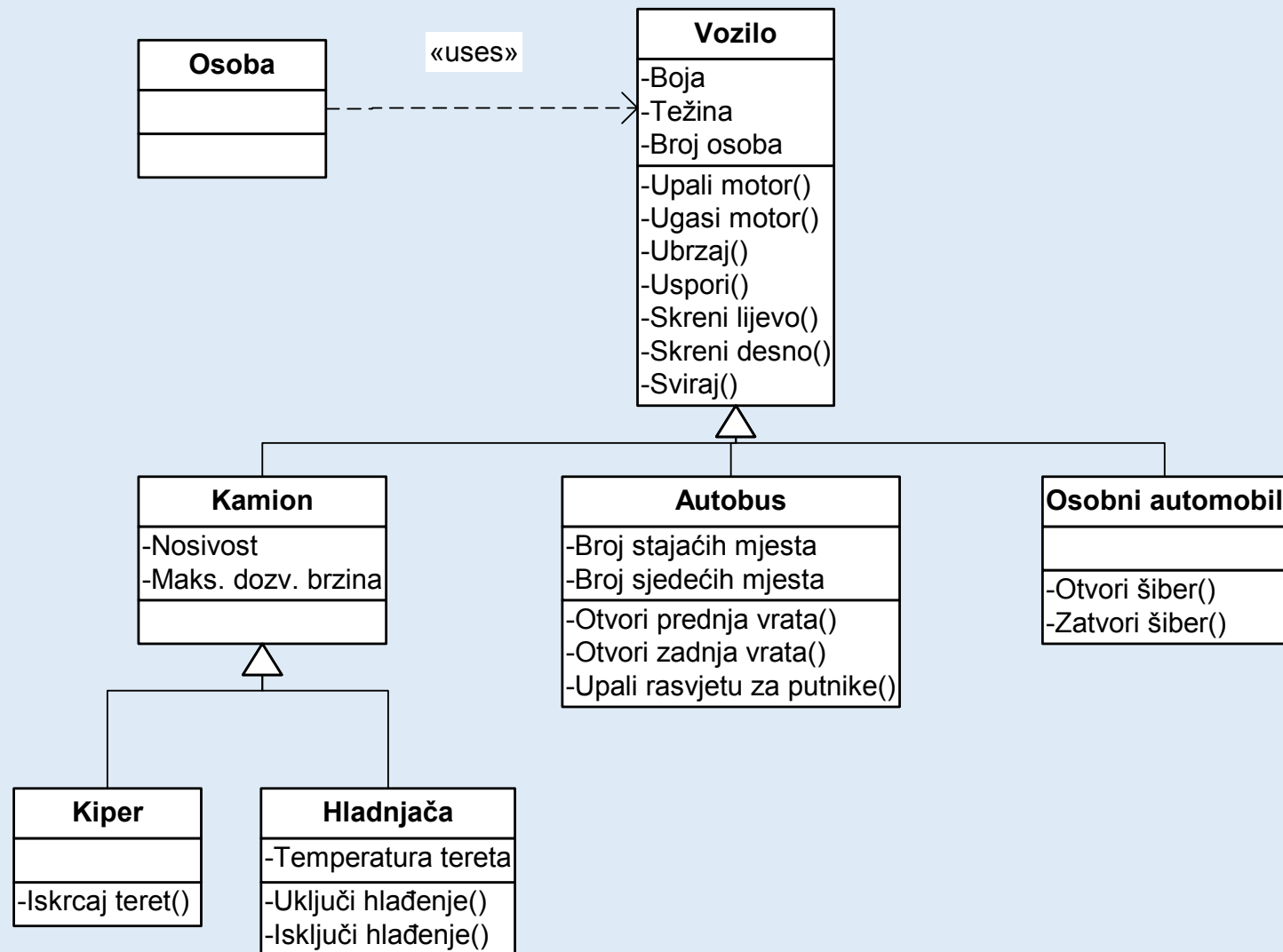
# Objektno programiranje

- Objektno programiranje je:
  - Tehnika programiranja u modernim programskim jezicima (Java, Visual Basic, C#, C++, ABAP)
- Objektno programiranje NIJE:
  - Povezano s objektima SAP repozitorija
  - Povezano sa poslovnim objektima (BOR) u SAP-u
  - Izrada i primjena BAPI-ja
  - Upotreba objekata (npr enjoySAP kontrola) na neobjektan način – česta zabluda neiskusnih programera u hibridnim jezicima

# Što su objekti?

- Dijelovi programa (slični varijablama) koji sadrže:
  - funkcionalnosti (tj. aktivnosti, operacije)
  - svojstva (tj. podatke, attribute, varijable, druge objekte...)
- Oponašaju karakteristike objekata “stvarnog svijeta”
- S vanjskim svijetom komuniciraju preko sučelja
- Najvažnije karakteristike:
  - Nasljeđivanje svojstava i funkcionalnosti
  - Potpuna odgovornost za svoju funkcionalnost

# Primjer: vozila



# Sadržaj

- Objektno programiranje i objekti
- • **Prednosti objektnog programiranja**
- ABAP kao objektni jezik
- Podizanje kompetencija
- Uzroci potencijalnog neuspjeha
- Uzorci dizajna (Design Patterns)
- Stvarni primjer primjene
- Zaključak

# Prednosti objektnog programiranja

- Jedan problem se rješava samo na jednom mjestu (točno određena raspodjela odgovornosti)
- Smanjenje koda specifičnog za aplikaciju
- Brz i lagan razvoj
- Jasna, razumljiva i intuitivna arhitektura
- Vuzualno modeliranje prije kodiranja (UML)
- Lagano održavanje
- Lagano prihvaćanje promijenjenih i novih zahtjeva
- Otvorenost za razvoj novih komponenti
- Konzistentnije i intuitivnije korisničko sučelje
- Oslobađanje resursa za kreativni rad

# Sadržaj

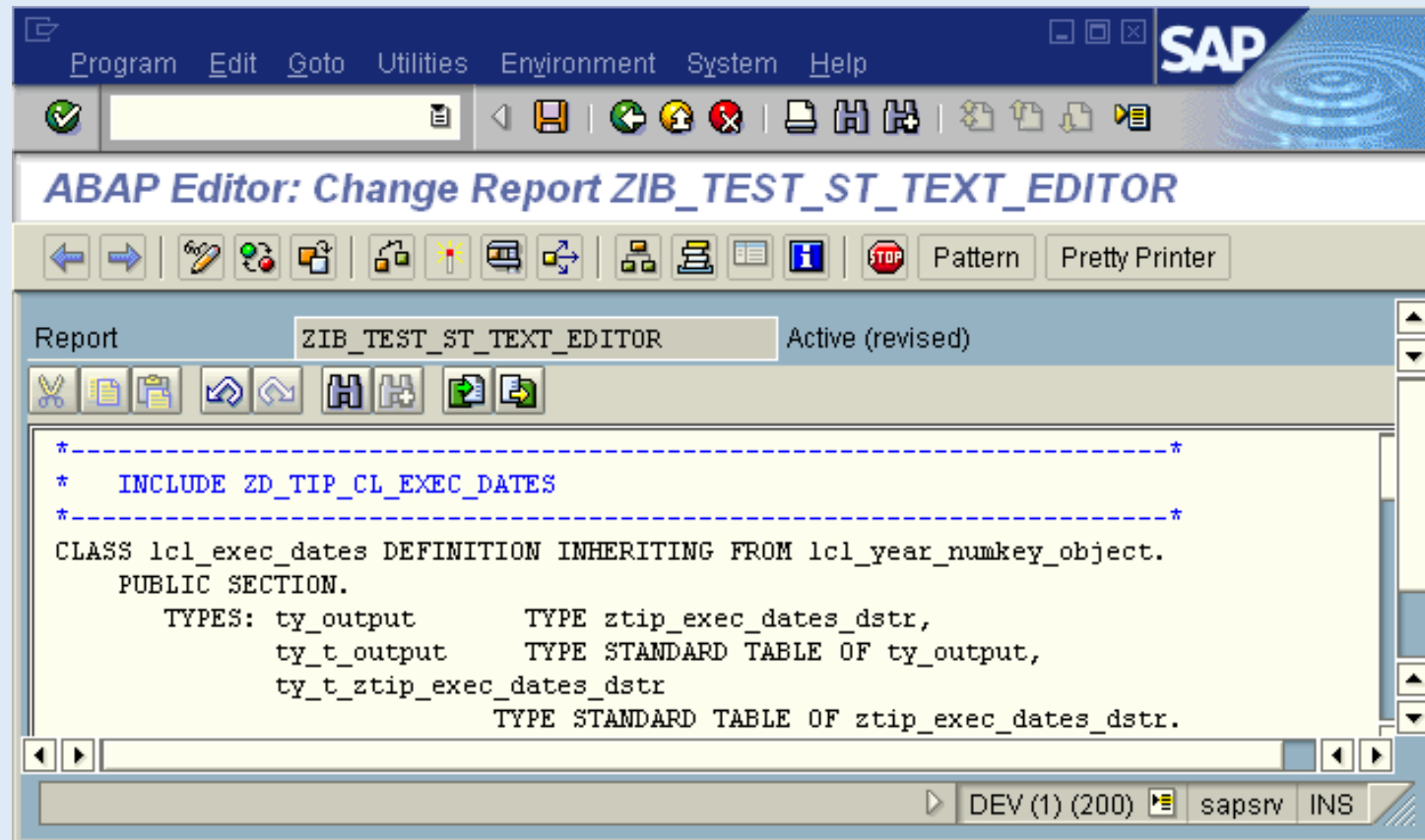
- Objektno programiranje i objekti
- Prednosti objektnog programiranja
- • **ABAP kao objektni jezik**
- Podizanje kompetencija
- Uzroci potencijalnog neuspjeha
- Uzorci dizajna (Design Patterns)
- Stvarni primjer primjene
- Zaključak

## Je li ABAP strogo objektno?

- **Nije**, on je hibridni jezik (kao npr Visual Basic)
- Skupo i nepotrebno preprogramirati cijeli SAP sustav u objekte
  - Kompatibilnost unazad
  - Hibridni jezik

# Alat za objektni ABAP

- Standardni ABAP editor



# Sadržaj

- Objektno programiranje i objekti
- Prednosti objektnog programiranja
- ABAP kao objektni jezik
- • **Podizanje kompetencija**
- Uzroci potencijalnog neuspjeha
- Uzorci dizajna (Design Patterns)
- Stvarni primjer primjene
- Zaključak

# Da li je potrebno podizanje kompetencija?

- Developeri i sistem-analitičari **da**
- Funkcionalni konzultanti **ne!**
  - Brinu za krajnji rezultat
  - Ne zanima ih što je “ispod haube”

# Sadržaj

- Objektno programiranje i objekti
- Prednosti objektnog programiranja
- ABAP kao objektni jezik
- Podizanje kompetencija
- • **Uzroci potencijalnog neuspjeha**
- Uzorci dizajna (Design Patterns)
- Stvarni primjer primjene
- Zaključak

# Uzroci potencijalnog neuspjeha

- Nemotiviranost programera da uče
- Neprimjenjivanje naučenog (u početku je stari način uvijek lakši)
- Neuviđanje prednosti - dolaze do izražaja više kod velikih aplikacija
- Obeshrabrenje zbog početnog neuspjeha

# Sadržaj

- Objektno programiranje i objekti
- Prednosti objektnog programiranja
- ABAP kao objektni jezik
- Podizanje kompetencija
- Uzroci potencijalnog neuspjeha
- • **Uzorci dizajna (Design Patterns)**
- Stvarni primjer primjene
- Zaključak

# Što su uzorci dizajna (*Design Patterns*)?

- Uzorci su provjerena i imenovana rješenja problema koji se ponavljaju u dizajnu
- U dizajnu objektnog software-a uzorci:
  - Pomažu u kreiranju fleksibilnih, razumljivih programa koji se lako održavaju i proširuju
  - Pomažu stručnjacima da lakše komuniciraju

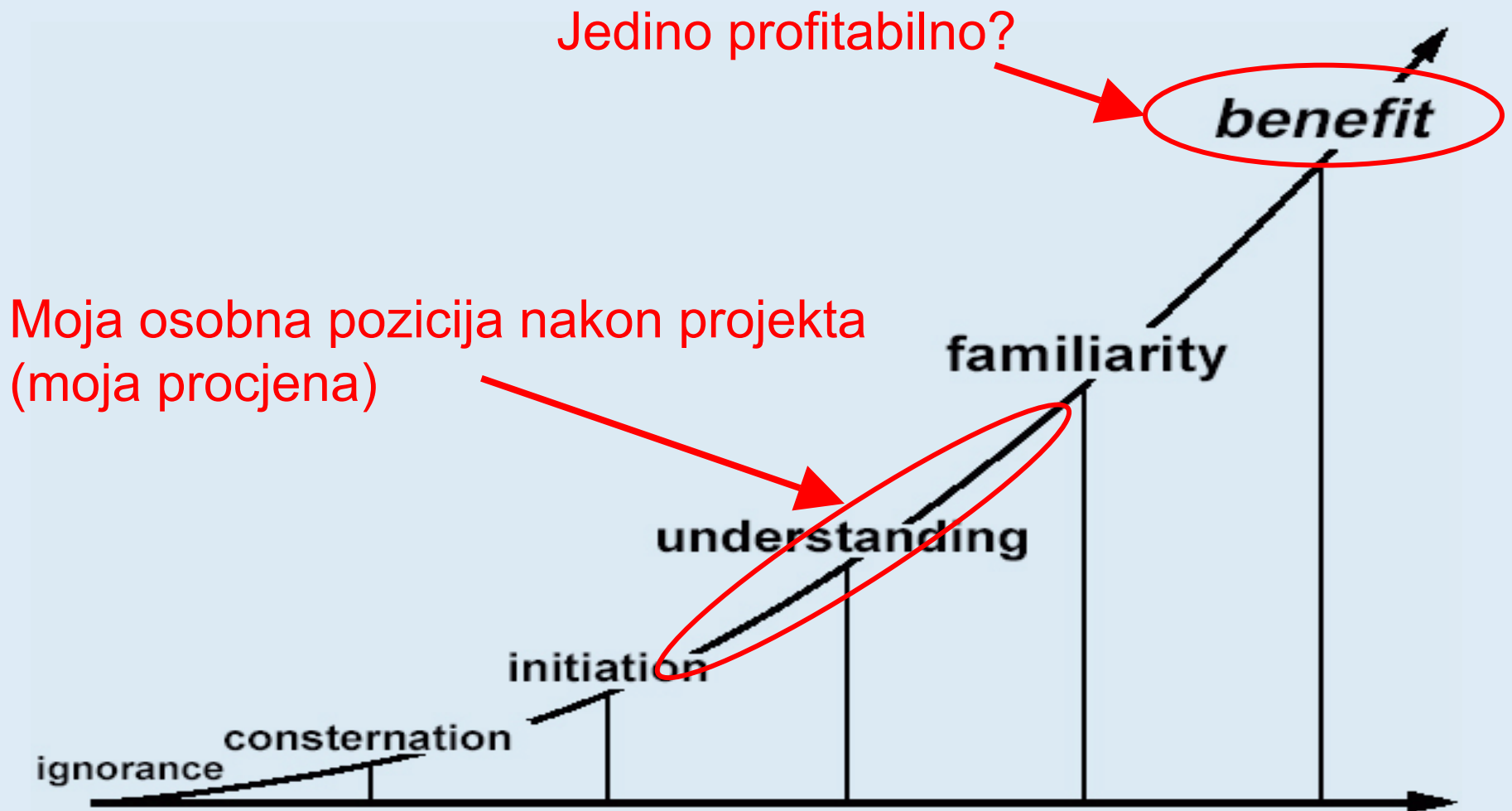
# Najvažnije pitanje

- Postoji konsenzus da su uzorci koristan koncept
- Ali da li su oni zaista... **profitabilni?**
- Stručnjaci nisu jednoglasni

# Pesimizam

- 90% svih developera nikada nije čulo za uzorke
- Teški su za naučiti
- Napor koji je potrebno uložiti je neprihvatljiv
- Ako niste majstor, gubite vrijeme
- Pogrešnom upotrebom više štete nego pomažu
- Važno je znati razliku između sličnih uzoraka

# Stupnjevi poznavanja uzoraka (John Vlissides)



# Alan Shalloway je optimističan!!!

- “...programeri koji su novi u objektno-orientiranom dizajnu mogu naučiti uzorke...”
- “...učenici koji su učili objektno-orientirani dizajn zajedno s uzorcima učili su objektno orientirani dizajn brže...”



Izvor: Shalloway/Trott – *Design Patterns Explained*

# Sadržaj

- Objektno programiranje i objekti
- Prednosti objektnog programiranja
- ABAP kao objektni jezik
- Podizanje kompetencija
- Uzroci potencijalnog neuspjeha
- Uzorci dizajna (Design Patterns)
- • **Stvarni primjer primjene**
- Zaključak

# Problem: nova velika aplikacija

- **Novi projekt:**  
Distribucija reklamnih listića u kućanstva
- **Moja pozicija prije projekta:**
  - Novi zahtjev vrlo kompliciran
  - Loše iskustvo s velikim aplikacijama u tradicionalnom pristupu
  - Početnik u objektnom dizajnu
  - Tek čuo za uzorke
  - Projekt odobren **s fleksibilnim vremenom**
  - Nisam još vidio pesimizam

# Ideja o okviru

## Napraviti okvir za izradu aplikacija:

- Rješiti ponavljajuće probleme u višestruko iskoristivim temeljnim komponentama
- Maksimizirati kod odvojen od aplikacije
- Minimizirati kod specifičan za aplikaciju
- Minimizirati ponavljajući manuelni rad (copy-paste)

# Sadržaj

- Objektno programiranje i objekti
- Prednosti objektnog programiranja
- ABAP kao objektni jezik
- Podizanje kompetencija
- Uzroci potencijalnog neuspjeha
- Uzorci dizajna (Design Patterns)
- Stvarni primjer primjene
- • **Zaključak**

# Zaključak

- Moja nova objektna ABAP aplikacija s uzorcima:
  - je intuitivna, lako razumljiva, konzistentna
  - radi pouzdano i robusno
  - lagano se održava i proširuje
- Nove aplikacije se razvijaju brzo i lako
- Temeljne komponente rade izvrsno u svim aplikacijama
- Objektni ABAP je puno snažniji od tradicionalnog (strukturiranog) programiranja
- Uzorci čine objektno ABAP još snažnijim
- Uzorci pomažu u komunikaciji između stručnjaka

# Pitanja i odgovori

# Kraj

Hvala i doviđenja!