

# Java 02 (objekty)

Luke McNemee

2014

# Objekt - motivace

- ▶ obvykle modelujeme svět

# Objekt - motivace

- ▶ obvykle modelujeme svět
- ▶ Svět je objektový

# Svět je objektový



Figure 1:

# Objekt

- ▶ Vše je objekt

# Objekt

- ▶ Vše je objekt
- ▶ Objekty komunikují pomocí zpráv

# Objekt

- ▶ Vše je objekt
- ▶ Objekty komunikují pomocí zpráv
- ▶ Objekty mají interní stav

# Objekt

- ▶ Vše je objekt
- ▶ Objekty komunikují pomocí zpráv
- ▶ Objekty mají interní stav
- ▶ Objekt je instancí třídy (class)



# Konstruktor

- ▶ pokud jej explicitně nenapíšeme, použije se bezparametrický (prázdný)

# Konstruktor

- ▶ pokud jej explicitně nenapíšeme, použije se bezparametrický (prázdný)
- ▶ můžeme definovat vlastní, jak s parametry, tak bezparametrický

# Konstruktor

- ▶ pokud jej explicitně nenapíšeme, použije se bezparametrický (prázdný)
- ▶ můžeme definovat vlastní, jak s parametry, tak bezparametrický
- ▶ použití pro iniciální nastavení vnitřního stavu objektu po vytvoření

# Zapouzdření

- ▶ public a private

# Zapouzdření

- ▶ public a private
- ▶ určuje viditelnost funkcí, proměnných pro okolí

# Zapouzdření

- ▶ public a private
- ▶ určuje viditelnost funkcí, proměnných pro okolí
- ▶ proměnné vždy private!!!

# Zapouzdření

- ▶ public a private
- ▶ určuje viditelnost funkcí, proměnných pro okolí
- ▶ proměnné vždy private!!!
- ▶ funkce dle potřeby

# Zapouzdření

- ▶ public a private
- ▶ určuje viditelnost funkcí, proměnných pro okolí
- ▶ proměnné vždy private!!!
- ▶ funkce dle potřeby
- ▶ slouží k omezení přístupu, tedy zabránění nechtěným zásahům do vnitřního stavu



# Zapouzdření

- ▶ public a private
- ▶ určuje viditelnost funkcí, proměnných pro okolí
- ▶ proměnné vždy private!!!
- ▶ funkce dle potřeby
- ▶ slouží k omezení přístupu, tedy zabránění nechtěným zásahům do vnitřního stavu
- ▶ z okolí je vidět pouze jednoduché rozhraní, uživatel nevidí implementaci

# Zapouzdření

- ▶ public a private
- ▶ určuje viditelnost funkcí, proměnných pro okolí
- ▶ proměnné vždy private!!!
- ▶ funkce dle potřeby
- ▶ slouží k omezení přístupu, tedy zabránění nechtěným zásahům do vnitřního stavu
- ▶ z okolí je vidět pouze jednoduché rozhraní, uživatel nevidí implementaci
- ▶ uživatel by měl být na implementaci nezávislý

# Detaily implementace

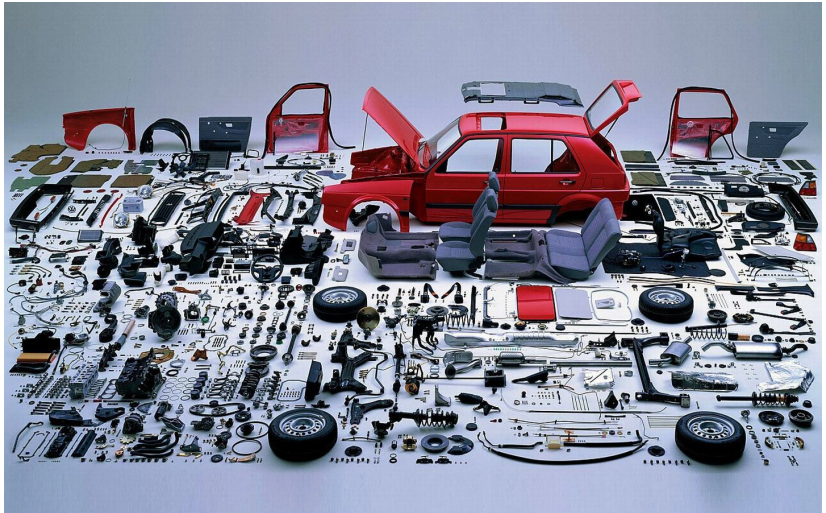


Figure 2:

# Proč zapouzdření?

- ▶ implementace není závislá na změnách okolního kódu

# Proč zapouzdření?

- ▶ implementace není závislá na změnách okolního kódu
- ▶ lepší ochrana vnitřního stavu proti chybám

# Proč zapouzdření?

- ▶ implementace není závislá na změnách okolního kódu
- ▶ lepší ochrana vnitřního stavu proti chybám
- ▶ kontrola nastavovaných parametrů

# Proč zapouzdření?

- ▶ implementace není závislá na změnách okolního kódu
- ▶ lepší ochrana vnitřního stavu proti chybám
- ▶ kontrola nastavovaných parametrů
- ▶ možnost změnit implementaci bez ohledu na uživatele

# Proč zapouzdření?

- ▶ implementace není závislá na změnách okolního kódu
- ▶ lepší ochrana vnitřního stavu proti chybám
- ▶ kontrola nastavovaných parametrů
- ▶ možnost změnit implementaci bez ohledu na uživatele
- ▶ navádí uživatele k správnému použití



# Static

- ▶ náleží třídě, nikoliv konkrétní instanci třídy

# Static

- ▶ náleží třídě, nikoliv konkrétní instanci třídy
- ▶ existuje pouze jedna instance statické položky

# Static

- ▶ náleží třídě, nikoliv konkrétní instanci třídy
- ▶ existuje pouze jedna instance statické položky
- ▶ typické využití pro univerzální funkce (připojení k databázi, převod do binárky ...)

# Static

- ▶ náleží třídě, nikoliv konkrétní instanci třídy
- ▶ existuje pouze jedna instance statické položky
- ▶ typické využití pro univerzální funkce (připojení k databázi, převod do binárky ...)
- ▶ statická funkce nemůže přistoupit k nestatickým funkcím stejného objektu (není svázaná se stejnou instancí)

# Semafor

- ▶ Zatím pouze binární (ukazuje 1 nebo 0)

# Semafor

- ▶ Zatím pouze binární (ukazuje 1 nebo 0)
- ▶ private proměnná se stavem

# Semafor

- ▶ Zatím pouze binární (ukazuje 1 nebo 0)
- ▶ private proměnná se stavem
- ▶ funkce pro přepnutí stavu

# Semafor

- ▶ Zatím pouze binární (ukazuje 1 nebo 0)
- ▶ private proměnná se stavem
- ▶ funkce pro přepnutí stavu
- ▶ funkce pro zjištění stavu



# Semafor

- ▶ Zatím pouze binární (ukazuje 1 nebo 0)
- ▶ private proměnná se stavem
- ▶ funkce pro přepnutí stavu
- ▶ funkce pro zjištění stavu
- ▶ funkce pro výpis informací

