DELTA TopGun

(16) Úvod do objektově orientovaného programování

Luboš Zápotočný, Tomáš Faltejsek, Michal Havelka

2023

Obsah

Programovací paradigmata

Programovací paradigmata

Programovací jazyky je možné seskupit podle vlastností, které podporují

Programovací paradigmata

Programovací jazyky je možné seskupit podle vlastností, které podporují

Těmto skupinám říkáme programovací paradigmata

Programovací paradigmata

Programovací jazyky je možné seskupit podle vlastností, které podporují

Těmto skupinám říkáme programovací paradigmata

Programovací jazyk může podporovat několik stylů programování a některé jazyky se tedy mohou vyskytovat ve více skupinách

Programovací paradigmata

Jaké programovací paradigmata znáte?

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)
- Objektově orientované

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)
- Objektově orientované (instrukce lze seskupit společně s nějakým stavem)

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)
- Objektově orientované (instrukce lze seskupit společně s nějakým stavem)

Deklarativní

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)
- Objektově orientované (instrukce lze seskupit společně s nějakým stavem)

Deklarativní (popisuje, jak by měl vypadat výsledek, ale nepopisuje postup výpočtu)

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)
- Objektově orientované (instrukce lze seskupit společně s nějakým stavem)

Deklarativní (popisuje, jak by měl vypadat výsledek, ale nepopisuje postup výpočtu)

- Funkcionální

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)
- Objektově orientované (instrukce lze seskupit společně s nějakým stavem)

Deklarativní (popisuje, jak by měl vypadat výsledek, ale nepopisuje postup výpočtu)

- Funkcionální (výsledek je definován aplikací posloupnosti funkcí)

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)
- Objektově orientované (instrukce lze seskupit společně s nějakým stavem)

Deklarativní (popisuje, jak by měl vypadat výsledek, ale nepopisuje postup výpočtu)

- Funkcionální (výsledek je definován aplikací posloupnosti funkcí)
- Logické

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)
- Objektově orientované (instrukce lze seskupit společně s nějakým stavem)

Deklarativní (popisuje, jak by měl vypadat výsledek, ale nepopisuje postup výpočtu)

- Funkcionální (výsledek je definován aplikací posloupnosti funkcí)
- Logické (definujeme fakta, pravidla a relace, výsledek je definován odpovědí na správně položenou otázku)

Programovací paradigmata

Jaké programovací paradigmata znáte?

Imperativní (přesně popisuje, jak se mění stav stroje)

- Procedurální (instrukce lze seskupit do procedur)
- Objektově orientované (instrukce lze seskupit společně s nějakým stavem)

Deklarativní (popisuje, jak by měl vypadat výsledek, ale nepopisuje postup výpočtu)

- Funkcionální (výsledek je definován aplikací posloupnosti funkcí)
- Logické (definujeme fakta, pravidla a relace, výsledek je definován odpovědí na správně položenou otázku)
- Reaktivní

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala (imperativní - objektově orientovaný)

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala (imperativní - objektově orientovaný)
- Prolog

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala (imperativní - objektově orientovaný)
- Prolog (deklarativní - logický)

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala (imperativní - objektově orientovaný)
- Prolog (deklarativní - logický)
- PHP

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala (imperativní - objektově orientovaný)
- Prolog (deklarativní - logický)
- PHP (imperativní - objektově orientovaný)

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala (imperativní - objektově orientovaný)
- Prolog (deklarativní - logický)
- PHP (imperativní - objektově orientovaný)
- Haskel

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala (imperativní - objektově orientovaný)
- Prolog (deklarativní - logický)
- PHP (imperativní - objektově orientovaný)
- Haskel (deklarativní - funkcionální)

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala (imperativní - objektově orientovaný)
- Prolog (deklarativní - logický)
- PHP (imperativní - objektově orientovaný)
- Haskel (deklarativní - funkcionální)
- Verilog

Programovací paradigmata - příklady

Kam byste zařadili jednotlivé jazyky?

- C (imperativní - procedurální)
- C++ (imperativní - objektově orientovaný)
- Java (imperativní - objektově orientovaný)
- Lisp (deklarativní - funkcionální)
- Go (imperativní - objektově orientovaný)
- Scala (imperativní - objektově orientovaný)
- Prolog (deklarativní - logický)
- PHP (imperativní - objektově orientovaný)
- Haskel (deklarativní - funkcionální)
- Verilog (deklarativní - reaktivní)

Objektově orientované jazyky lze většinou použít i bez objektů, řadí se tedy i do skupiny procedurálních jazyků

Čtyři pilíře objektově orientovaného programování

Čtyři pilíře objektově orientovaného programování

- abstraction - abstrakce

Čtyři pilíře objektově orientovaného programování

- abstraction - abstrakce
- encapsulation - zapouzdření

Čtyři pilíře objektově orientovaného programování

- abstraction - abstrakce
- encapsulation - zapouzdření
- inheritance - dědičnost

Čtyři pilíře objektově orientovaného programování

- abstraction - abstrakce
- encapsulation - zapouzdření
- inheritance - dědičnost
- polymorphism - polymorfismus

Abstrakce

Snažíme se redukovat složitost programu tak, že nepotřebné věci skrýváme

Abstrakce

Snažíme se redukovat složitost programu tak, že nepotřebné věci skrýváme

Uživatel může vytvářen komplexní řešení bez znalosti našeho komplexního řešení

Abstrakce

Snažíme se redukovat složitost programu tak, že nepotřebné věci skrýváme

Uživatel může vytvářen komplexní řešení bez znalosti našeho komplexního řešení

Příklad

vytváříme abstraktní vrstvu mezi

Abstrakce

Snažíme se redukovat složitost programu tak, že nepotřebné věci skrýváme

Uživatel může vytvářen komplexní řešení bez znalosti našeho komplexního řešení

Příklad

vytváříme abstraktní vrstvu mezi

- aplikační logikou a přístupem do databáze

Abstrakce

Snažíme se redukovat složitost programu tak, že nepotřebné věci skrýváme

Uživatel může vytvářen komplexní řešení bez znalosti našeho komplexního řešení

Příklad

vytváříme abstraktní vrstvu mezi

- aplikační logikou a přístupem do databáze
- veřejným API a přístupem na lokální disk

Abstrakce

Snažíme se redukovat složitost programu tak, že nepotřebné věci skrýváme

Uživatel může vytvářen komplexní řešení bez znalosti našeho komplexního řešení

Příklad

vytváříme abstraktní vrstvu mezi

- aplikační logikou a přístupem do databáze
- veřejným API a přístupem na lokální disk
- veřejným REST API a jiným SOAP API

Zapouzdření

Komplexní objekt může zveřejňovat pro smysluplné (pro klienta) funkce či data

Zapouzdření

Komplexní objekt může zveřejňovat pro smysluplné (pro klienta) funkce či data

Objekt funguje jako černá skřínka, s klientem komunikuje pouze pomocí vstupních a výstupných dat, ale výpočet či jiné provedené akce jsou pro externího klienta skryté

Příklad

vytváříme abstraktní vrstvu mezi objektem člověk a pracovním portálem

Zapouzdření

Komplexní objekt může zveřejňovat pro smysluplné (pro klienta) funkce či data

Objekt funguje jako černá skřínka, s klientem komunikuje pouze pomocí vstupních a výstupných dat, ale výpočet či jiné provedené akce jsou pro externího klienta skryté

Příklad

vytváříme abstraktní vrstvu mezi objektem člověk a pracovním portálem

- objekt obsahuje privátní položku (seznam) pro uložení data nástupů a ukončení studií

Zapouzdření

Komplexní objekt může zveřejňovat pro smysluplné (pro klienta) funkce či data

Objekt funguje jako černá skřínka, s klientem komunikuje pouze pomocí vstupních a výstupných dat, ale výpočet či jiné provedené akce jsou pro externího klienta skryté

Příklad

vytváříme abstraktní vrstvu mezi objektem člověk a pracovním portálem

- objekt obsahuje privátní položku (seznam) pro uložení data nástupů a ukončení studií
- objekt poskytuje veřejnou metodu pro získání počtu roků, kdy daný člověk studoval

Zapouzdření

Komplexní objekt může zveřejňovat pro smysluplné (pro klienta) funkce či data

Objekt funguje jako černá skřínka, s klientem komunikuje pouze pomocí vstupních a výstupných dat, ale výpočet či jiné provedené akce jsou pro externího klienta skryté

Příklad

vytváříme abstraktní vrstvu mezi objektem člověk a pracovním portálem

- objekt obsahuje privátní položku (seznam) pro uložení data nástupů a ukončení studií
- objekt poskytuje veřejnou metodu pro získání počtu roků, kdy daný člověk studoval
- v rámci aplikační logiky tento výpočet může být velmi složitý (obsahuje nějaký právní řád, speciální pravidla či výjimky pro výpočet)

Dědičnost

Všichni lidé mají datum narození

Dědičnost

Všichni lidé mají datum narození

Všichni středoškoláci mají datum přijetí na střední školu

Dědičnost

Všichni lidé mají datum narození

Všichni středoškoláci mají datum přijetí na střední školu

Zároveň ale všichni středoškoláci jsou lidé a měli by tedy mít i datum narození

Dědičnost

Všichni lidé mají datum narození

Všichni středoškoláci mají datum přijetí na střední školu

Zároveň ale všichni středoškoláci jsou lidé a měli by tedy mít i datum narození

Na středoškoláka lze pohlížet jako na speciální podskupinu všech lidí

Dědičnost

Všichni lidé mají datum narození

Všichni středoškoláci mají datum přijetí na střední školu

Zároveň ale všichni středoškoláci jsou lidé a měli by tedy mít i datum narození

Na středoškoláka lze pohlížet jako na speciální podskupinu všech lidí

Pokud ale středoškolák přijde do nemocnice, nikoho už nezajímá, že je středoškolák a chtějí po něm pouze datum narození

Dědičnost

Všichni lidé mají datum narození

Všichni středoškoláci mají datum přijetí na střední školu

Zároveň ale všichni středoškoláci jsou lidé a měli by tedy mít i datum narození

Na středoškoláka lze pohlížet jako na speciální podskupinu všech lidí

Pokud ale středoškolák přijde do nemocnice, nikoho už nezajímá, že je středoškolák a chtějí po něm pouze datum narození

Středoškolák ale oproti normálním lidem umí vyřešit kvadratickou rovnici

Dědičnost

Všichni lidé mají datum narození

Všichni středoškoláci mají datum přijetí na střední školu

Zároveň ale všichni středoškoláci jsou lidé a měli by tedy mít i datum narození

Na středoškoláka lze pohlížet jako na speciální podskupinu všech lidí

Pokud ale středoškolák přijde do nemocnice, nikoho už nezajímá, že je středoškolák a chtějí po něm pouze datum narození

Středoškolák ale oproti normálním lidem umí vyřešit kvadratickou rovnici

Vysokoškolák by měl umět vše, co středoškolák, ale také musí umět hledat extrémy funkce více proměnných

Polymorfismus

Podstata polymorfismu spočívá v tom, že máme skupinu různorodých objektů, které ale mají společné schopnosti

Polymorfismus

Podstata polymorfismu spočívá v tom, že máme skupinu různorodých objektů, které ale mají společné schopnosti

Každý objekt ale danou schopnost umí dělat pouze svým způsobem

Polymorfismus

Podstata polymorfismu spočívá v tom, že máme skupinu různorodých objektů, které ale mají společné schopnosti

Každý objekt ale danou schopnost umí dělat pouze svým způsobem

Například abstraktní třída bude definovat rozhraní pro svou identifikaci
Člověk bude vracet své jméno, letadlo bude vracet číslo letu a kočka bude vracet "Mňouk"