Návrh – rozhraní a komponenty

Ing. Jiří Mlejnek

Katedra softwarového inženýrství
Fakulta informačních technologií
České vysoké učení technické v Praze

jiri.mlejnek@fit.cvut.cz
Softwarové inženýrství BI-SI1
ZS 2015/ Před. 7

Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

Obsah

- Rozhraní a komponenty
- Diagram nasazení
- GoF vzory

Rozhraní a komponenty

- Komponenta
  - Fyzická část systému
  - Samostatně nasaditelná
    - Soubor (exe, jar, war, dll)
  - Oddělená rozhraním od ostatních komponent
  - Má vnitřní strukturu

Rozhraní a komponenty

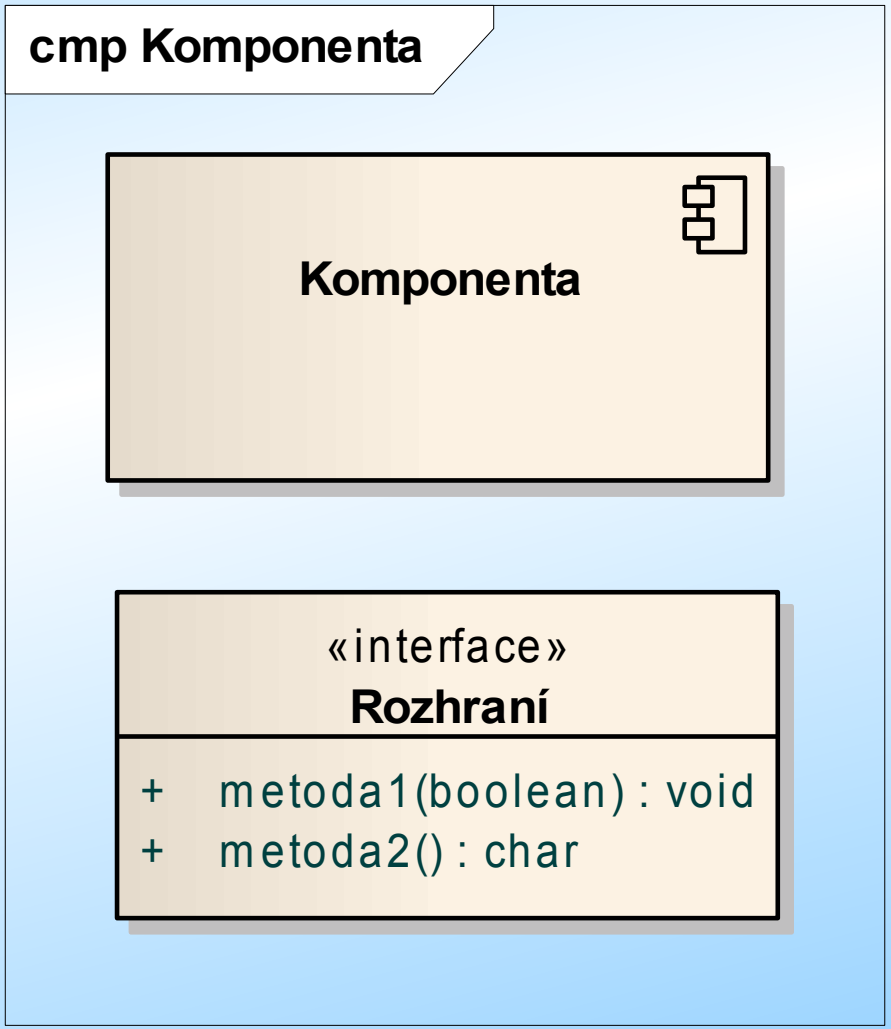
- Rozhraní
  - Množina operací
  - Odděluje specifikaci od implementace
  - Snižuje provázanost
  - Může být popsán různým způsobem
    - Interface programovacího jazyka (Java, .NET)
    - WSDL – popis rozhraní pro webové služby
    - Package – popis rozhraní pro PL/SQL

Rozhraní a komponenty

- Diagram komponent
  - Patří do skupiny diagramů struktur
  - Využití
    - Znázornění členění systému na komponenty
    - Zachycení rozhraní mezi komponentami
  - EA: UML Structural - Component

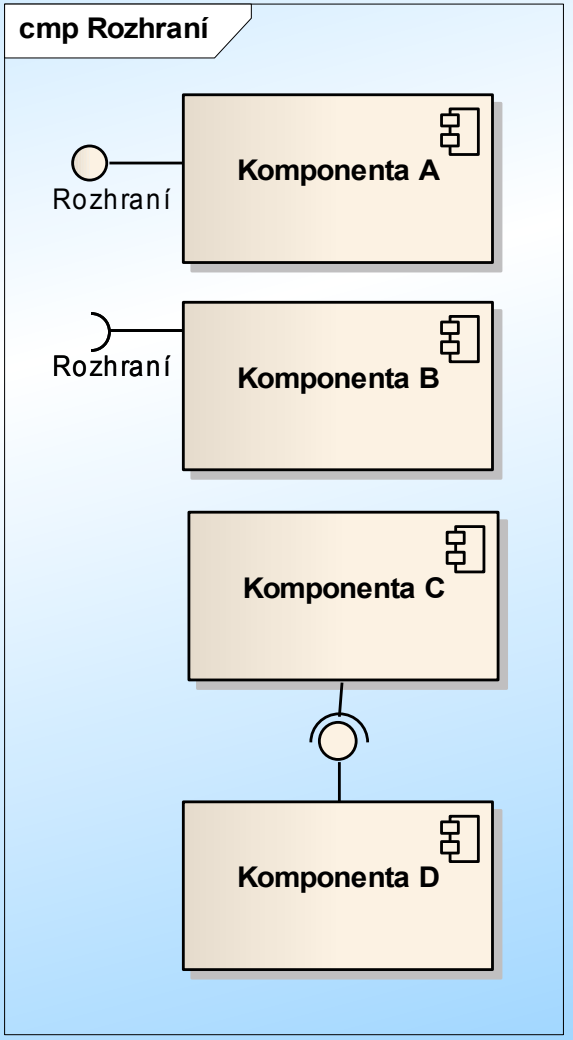
Rozhraní a komponenty

- Komponenta
- Rozhraní (Interface)



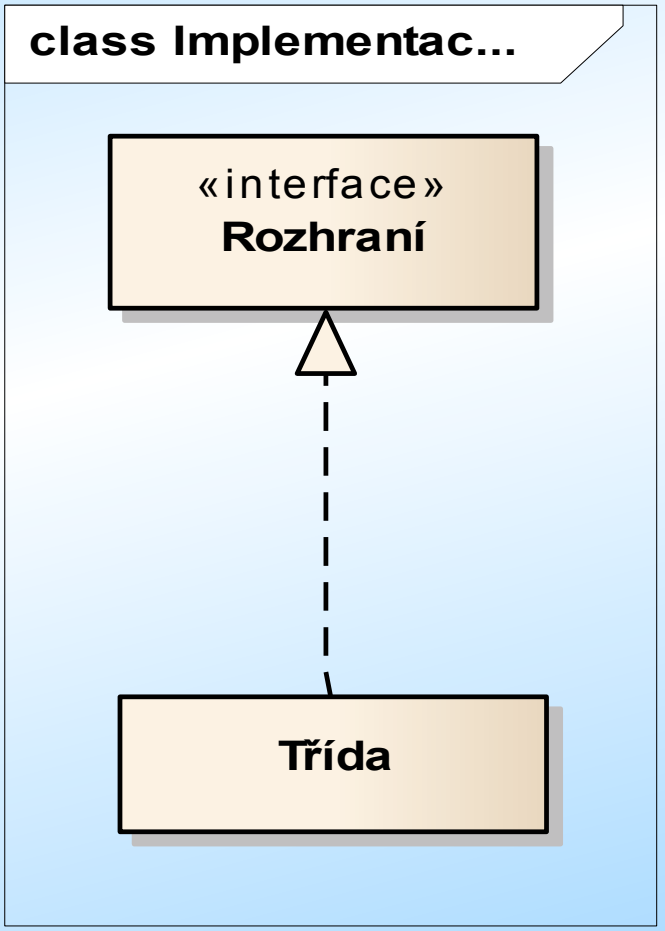
Rozhraní a komponenty

- Rozhraní (interface)
  - Nabízený
  - Vyžadovaný
- Sestavení

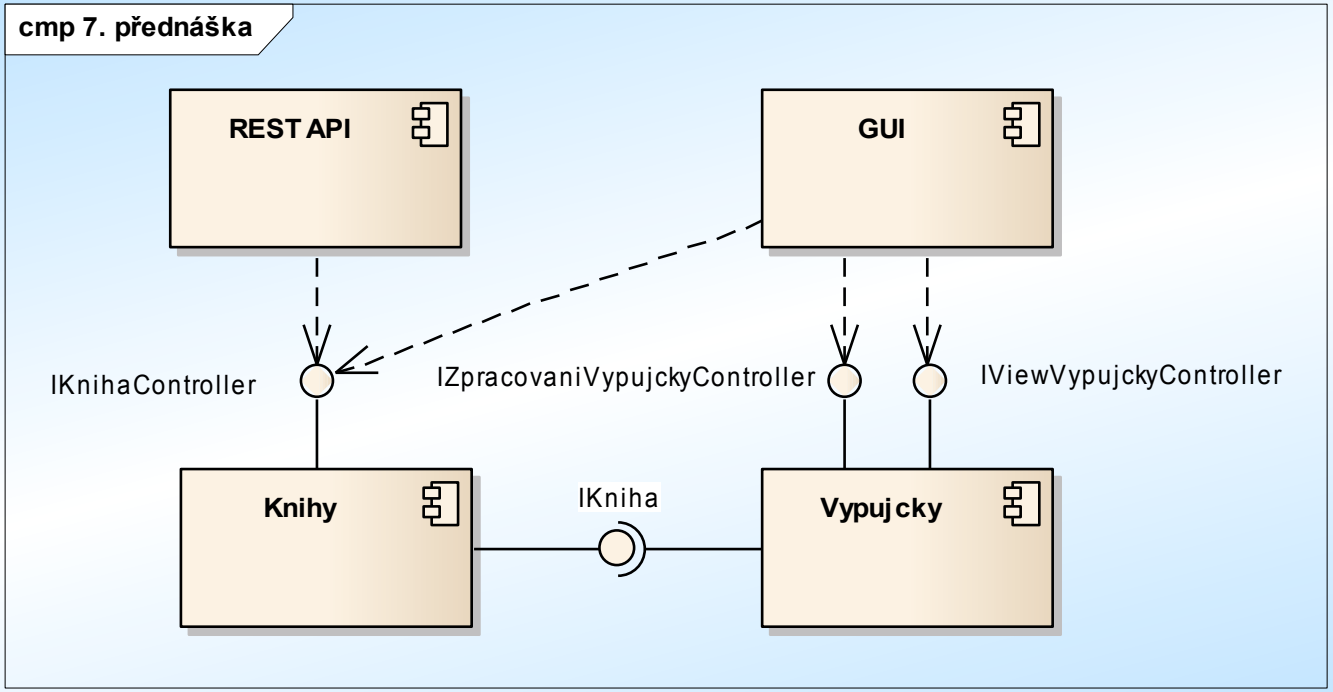


Rozhraní a komponenty

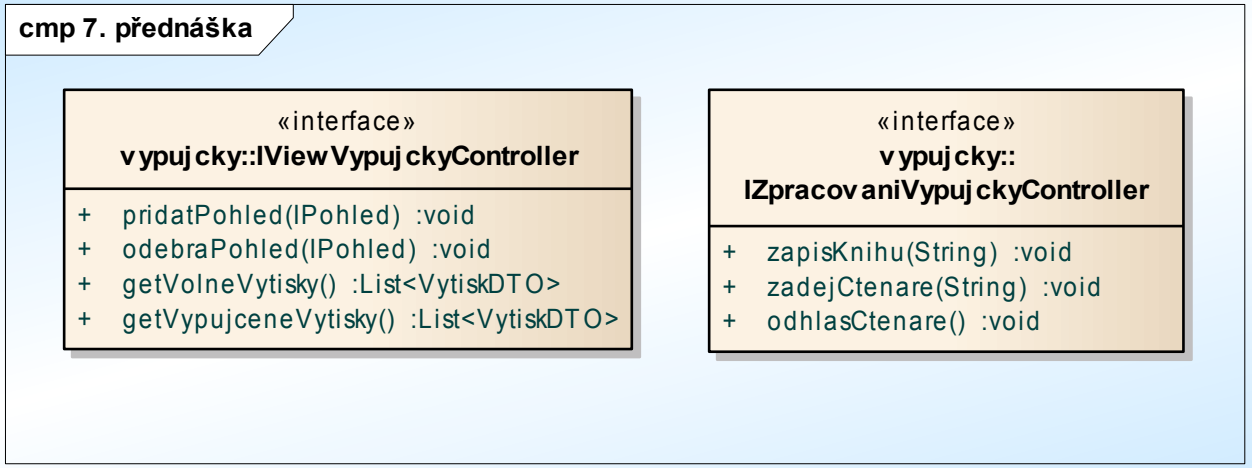
- Implementace rozhraní



Rozhraní a komponenty



Rozhraní a komponenty

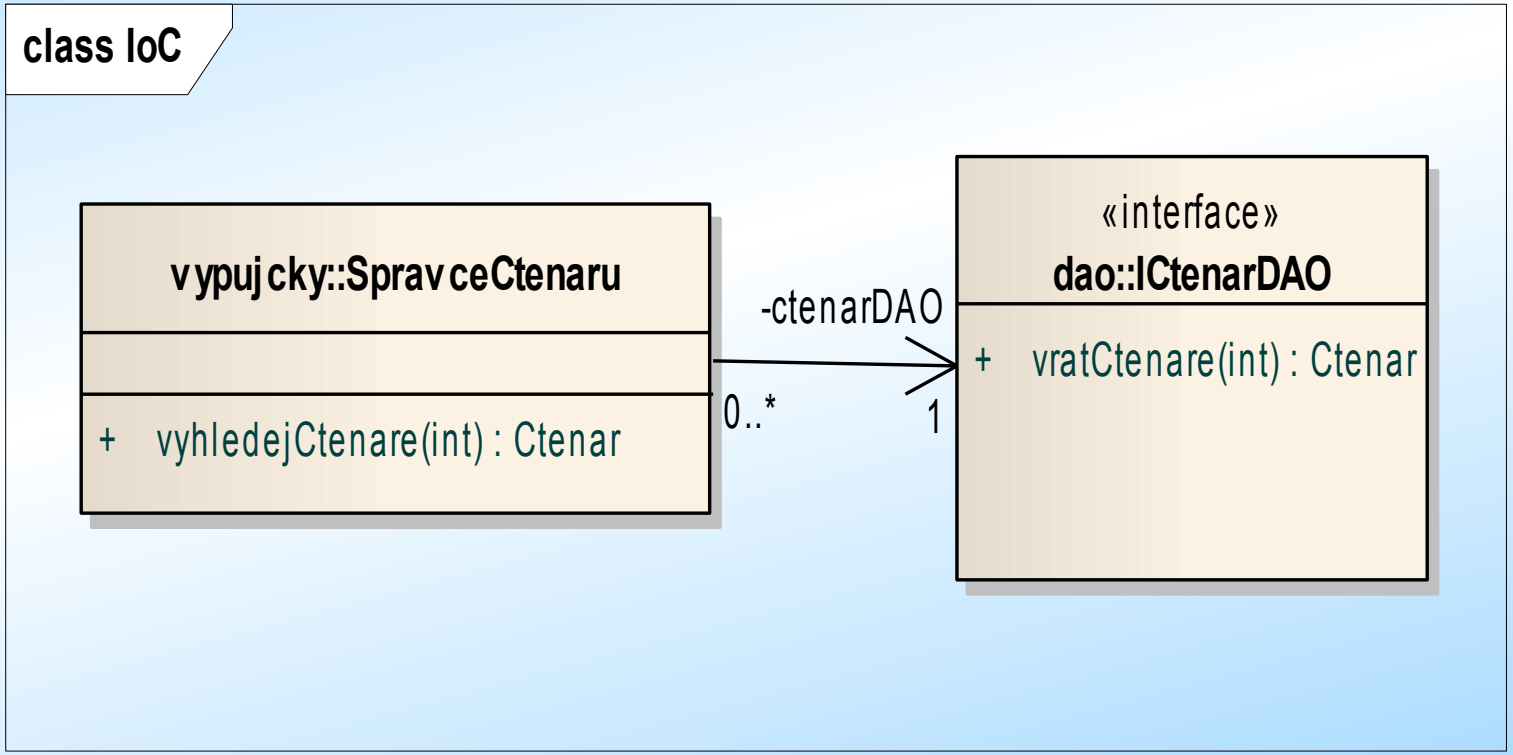


Rozhraní a komponenty

- Propojení komponent (tříd)
  - Manuálně ve zdrojovém kódu
  - IoC - Inversion of Control
    - Deklarativní vazby mezi třídami
    - Snadná výměna implementace
    - Existující řešení (IoC kontejnery)
      - EJB
      - Spring
      - Symfony (Service Configurator)

Rozhraní a komponenty

- Propojení pomocí rozhraní



Rozhraní a komponenty

```
public class SpravceCtenaru {
   private ICtenarDAO ctenarDAO = new CtenarDAO();

   public Ctenar vyhledejCtenare(int
cisloPrukazky){
      return ctenarDAO.vratCtenare(cisloPrukazky);
   }
}
```

Rozhraní a komponenty

```
public class SpravceCtenaru {
   private ICtenarDAO ctenarDAO;

   public Ctenar vyhledejCtenare(int cisloPrukazky{
      return ctenarDAO.vratCtenare(cisloPrukazky);
   }
}
```

Rozhraní a komponenty

```
<bean name="spravceCtenaru"
class="knihovna.bl.vypujcky.SpravceCtenaru">
  <property name="ctenarDAO">
    <ref bean="ctenarDAO"/>
  </property>
</bean>

<bean name="ctenarDAO"
class="knihovna.dl.dao.hibernate.CtenarDAO">
</bean>
```

Rozhraní a komponenty

- Spring
  - Scope
    - Singleton
    - Prototype
    - Request
    - Session
  - DI
    - Constructor-based
    - Setter-based

Rozhraní a komponenty

Dotazy?

Diagram nasazení

- Popisuje fyzickou architekturu
  - Rozložení komponent na výpočetní uzly
- Příklady
  - Klient – server
    - Tlustý klient
    - Tenký klient
    - Aplikační cluster
  - P2P
- EA: UML Structural - Deployment

Diagram nasazení

- Využití
  - Popis fyzického rozmístění komponent
  - Popis fyzických zařízení a jejich propojení
  - Instalační příručka

Diagram nasazení

- Uzel (node)

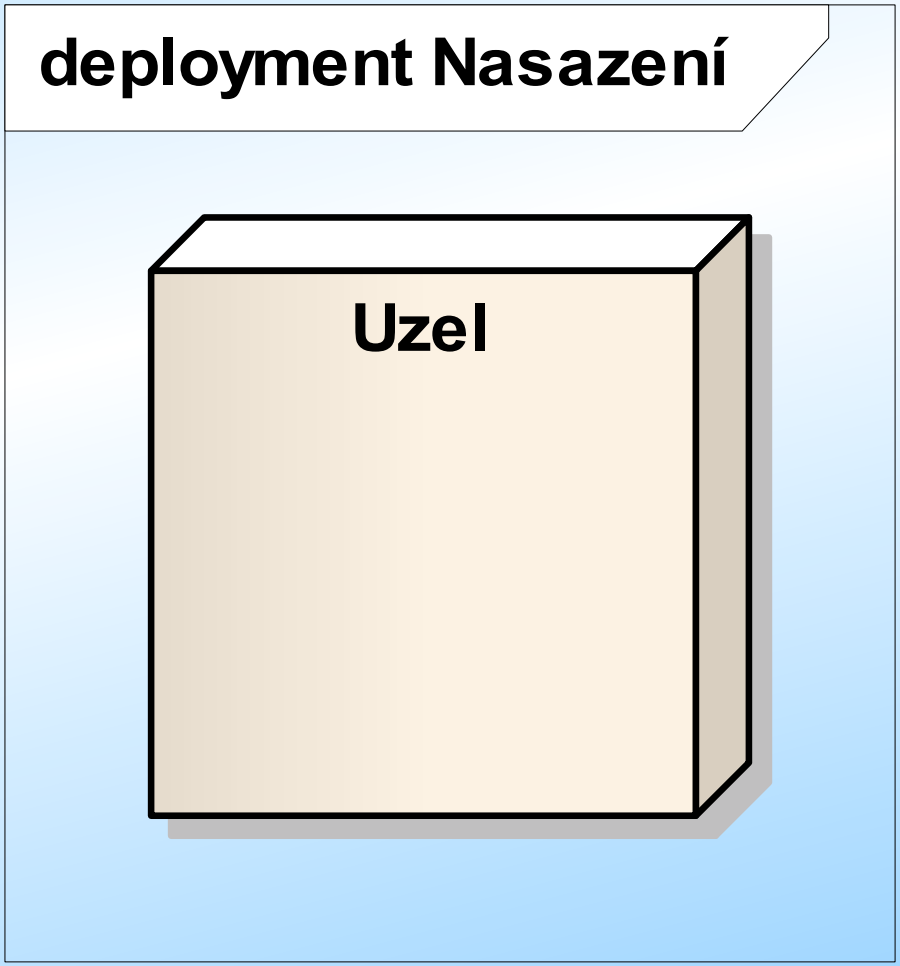


Diagram nasazení

- Stereotypy
  - Execution environment
    - Tomcat
    - Apache
    - Firefox
  - Device
    - Server
    - Notebook
    - PC
    - Tiskárna

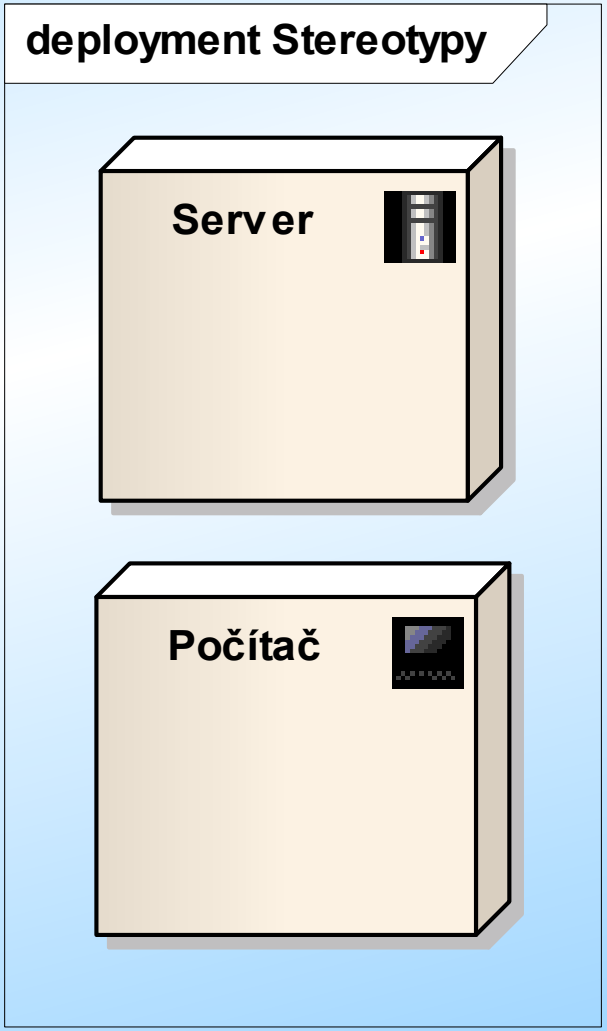


Diagram nasazení

- Desktopová aplikace

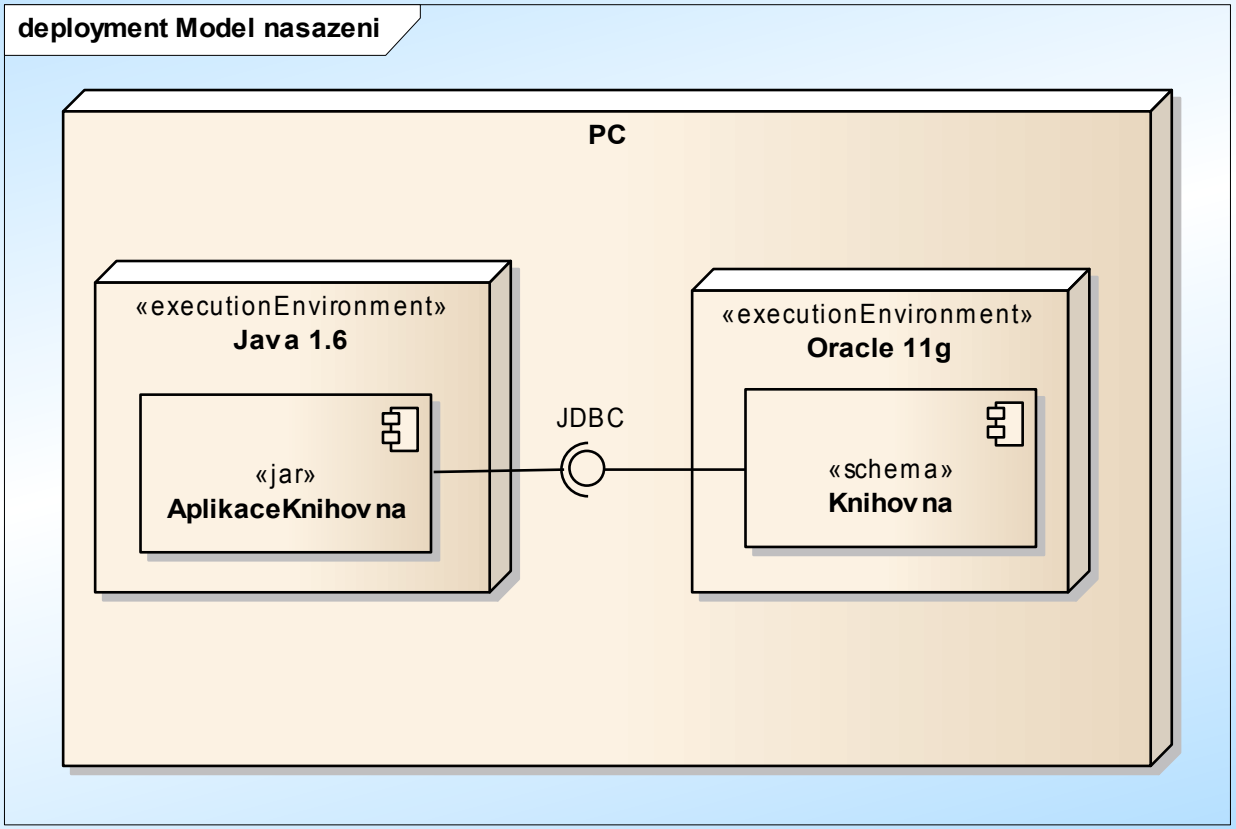


Diagram nasazení

- Klient - server

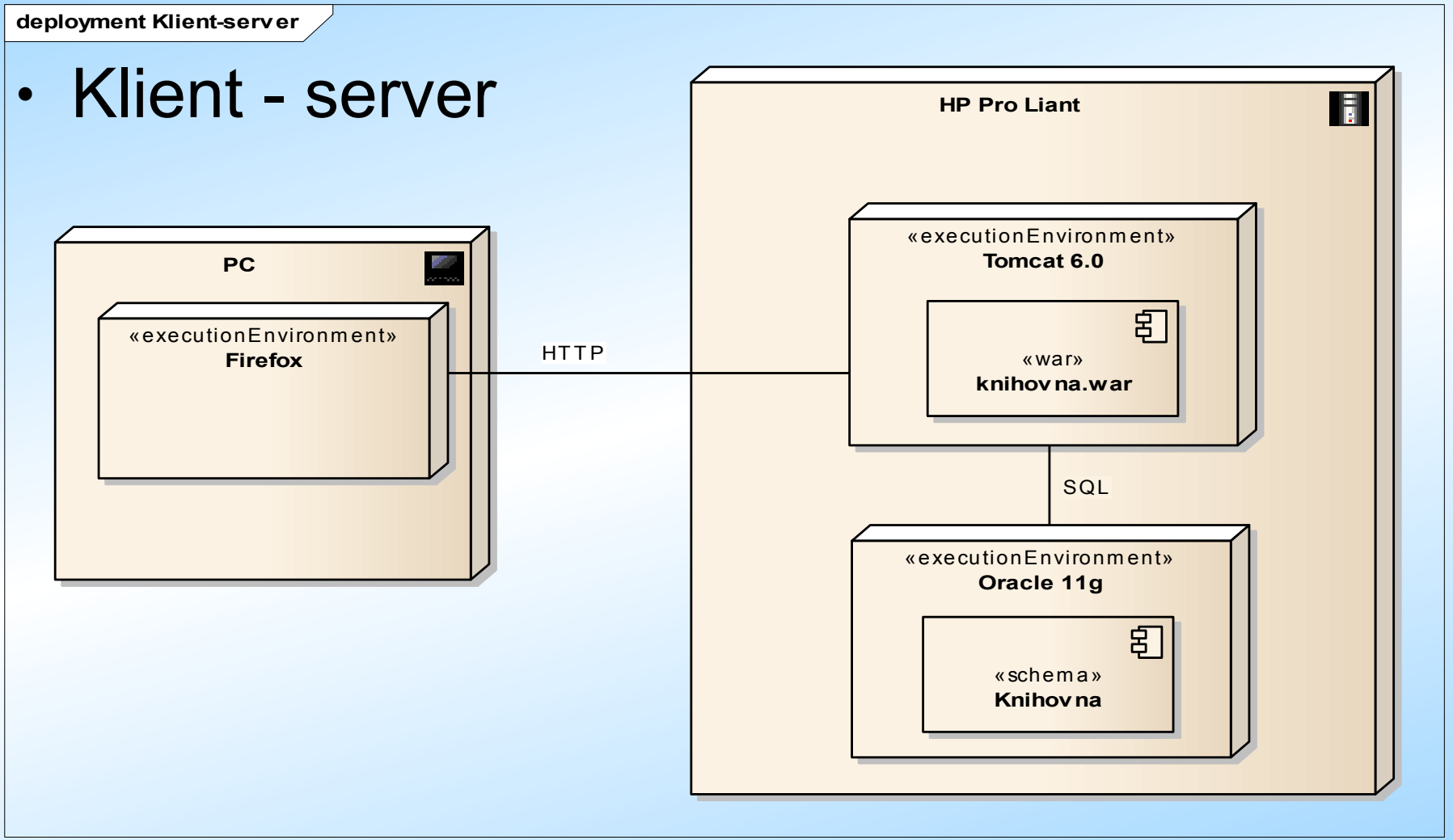


Diagram nasazení

Dotazy?

GoF vzory

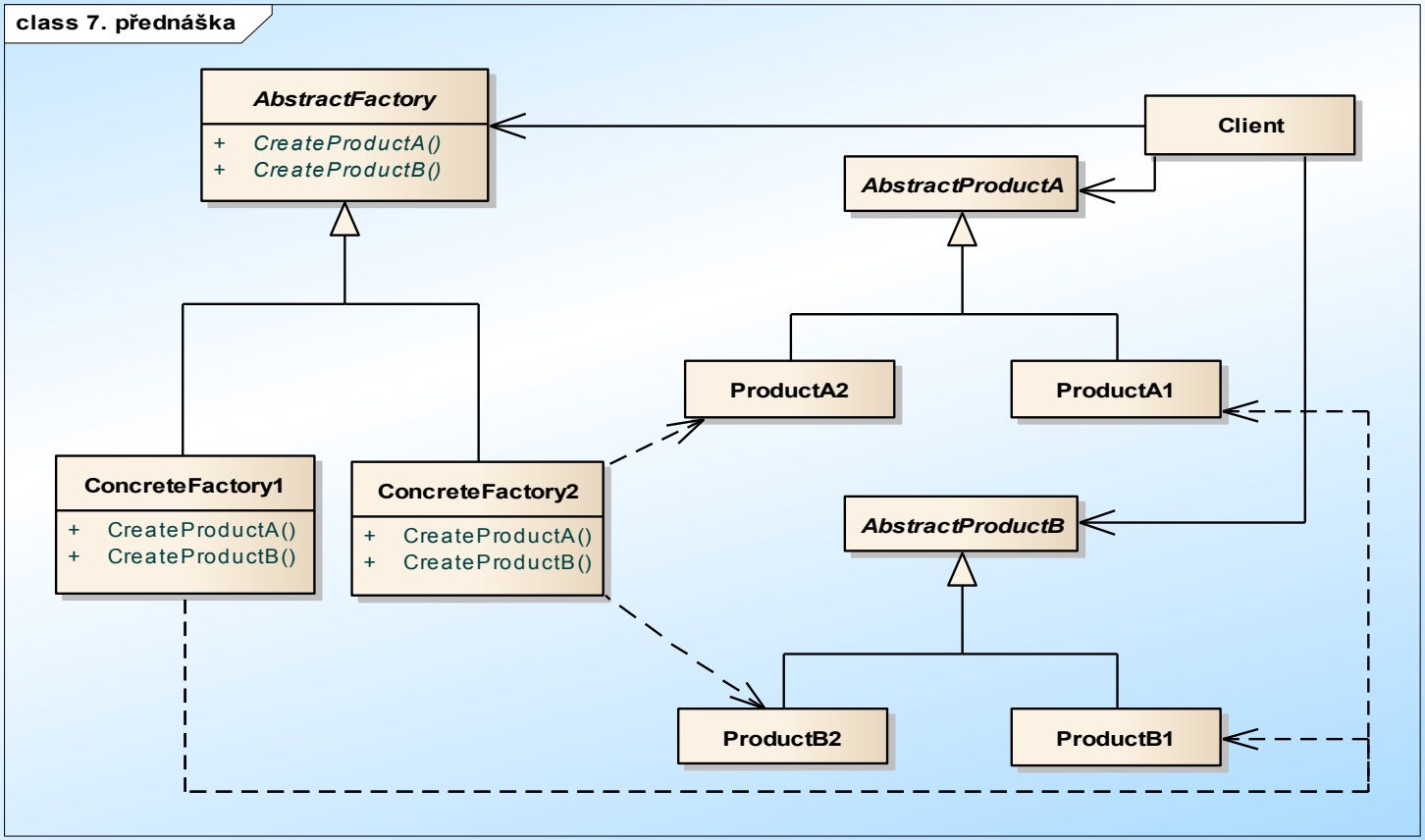
- Rozdělení
  - Vzory pro vytváření objektů (Creational)
    - Abstraktní továrna (Abstract Factory)
    - Jedináček (Singleton)
  - Strukturální vzory (Structural)
    - Adaptér (Adapter)
    - Fasáda (Facade)
  - Vzory chování (Behavioral)
    - Stav (State)
    - Pozorovatel (Observer)

GoF vzory

- Abstraktní továrna (Abstract Factory)
  - Poskytuje rozhraní pro vytváření skupiny objektů bez znalosti konkrétní implementace
  - Konkrétní implementace je dána použitou továrnou

GoF vzory

- Abstraktní továrna (Abstract Factory)



GoF vzory

- Abstraktní továrna (Abstract Factory)

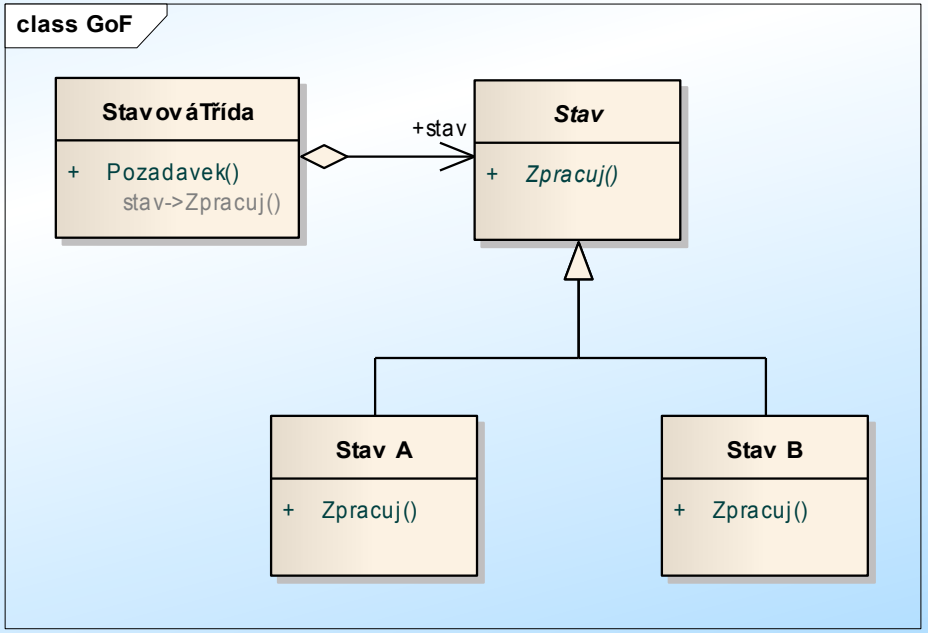
```
public class SpravceCtenaru {
    private FactoryDAO factory = new TestFactoryDAO();

    private ICtenarDAO ctenarDAO = factory.getCtenarDAO();


    private IVypDAO vypujckaDAO = factory.getVypujckaDAO();

    public Ctenar vyhledejCtenare(int cisloPrukazky){
       return ctenarDAO.vratCtenare(cisloPrukazky);
    }
}
```

GoF vzory

- Stav (State)
  - Odděluje chování třídy závislé na stavu do samostatné třídy
  - Odstraňuje složitá větvení (switch, case, if, else)

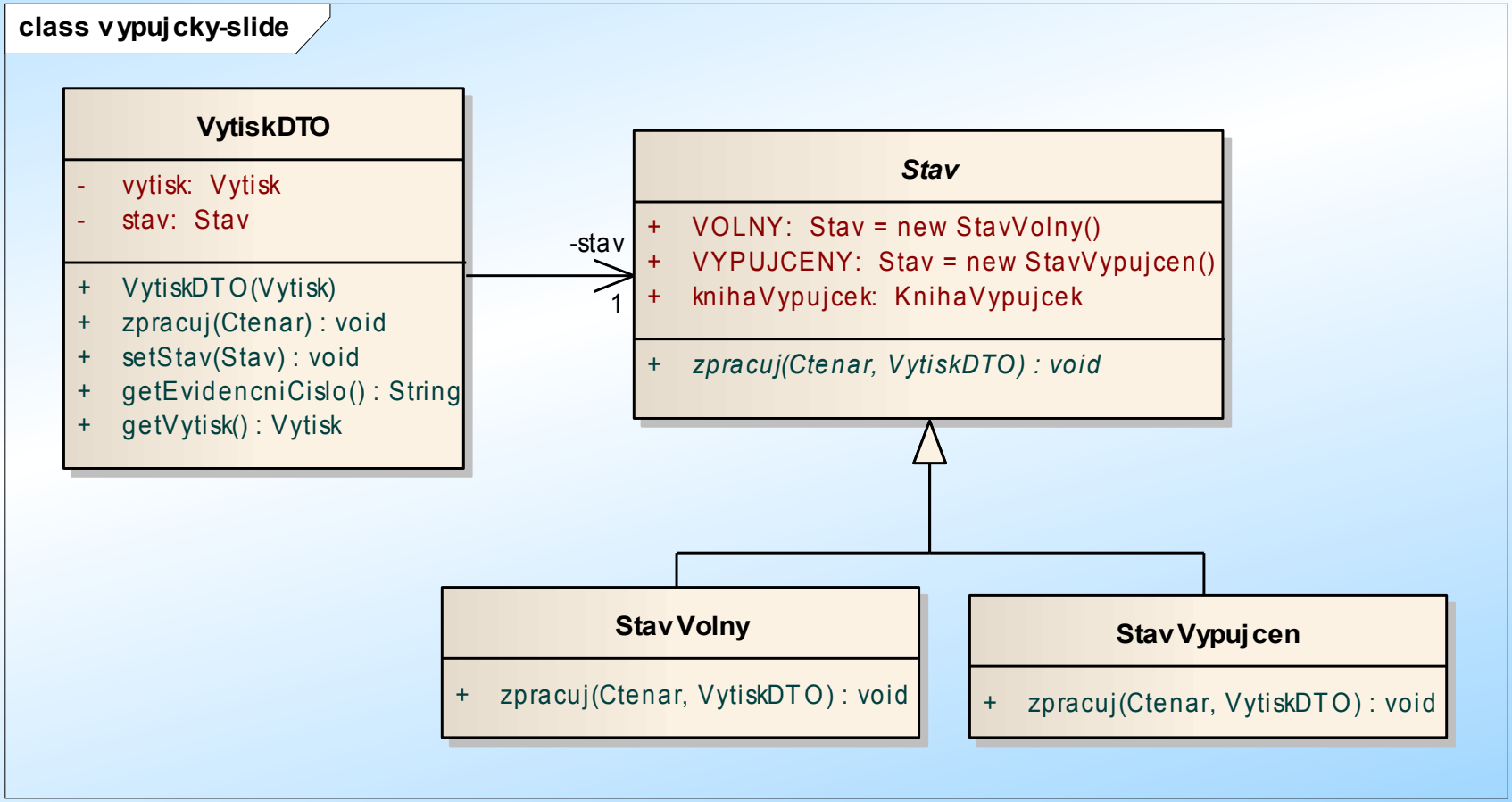
GoF vzory

- Stav (State)
  - Libovolný počet stavů
  - Snadné přidání nového stavu



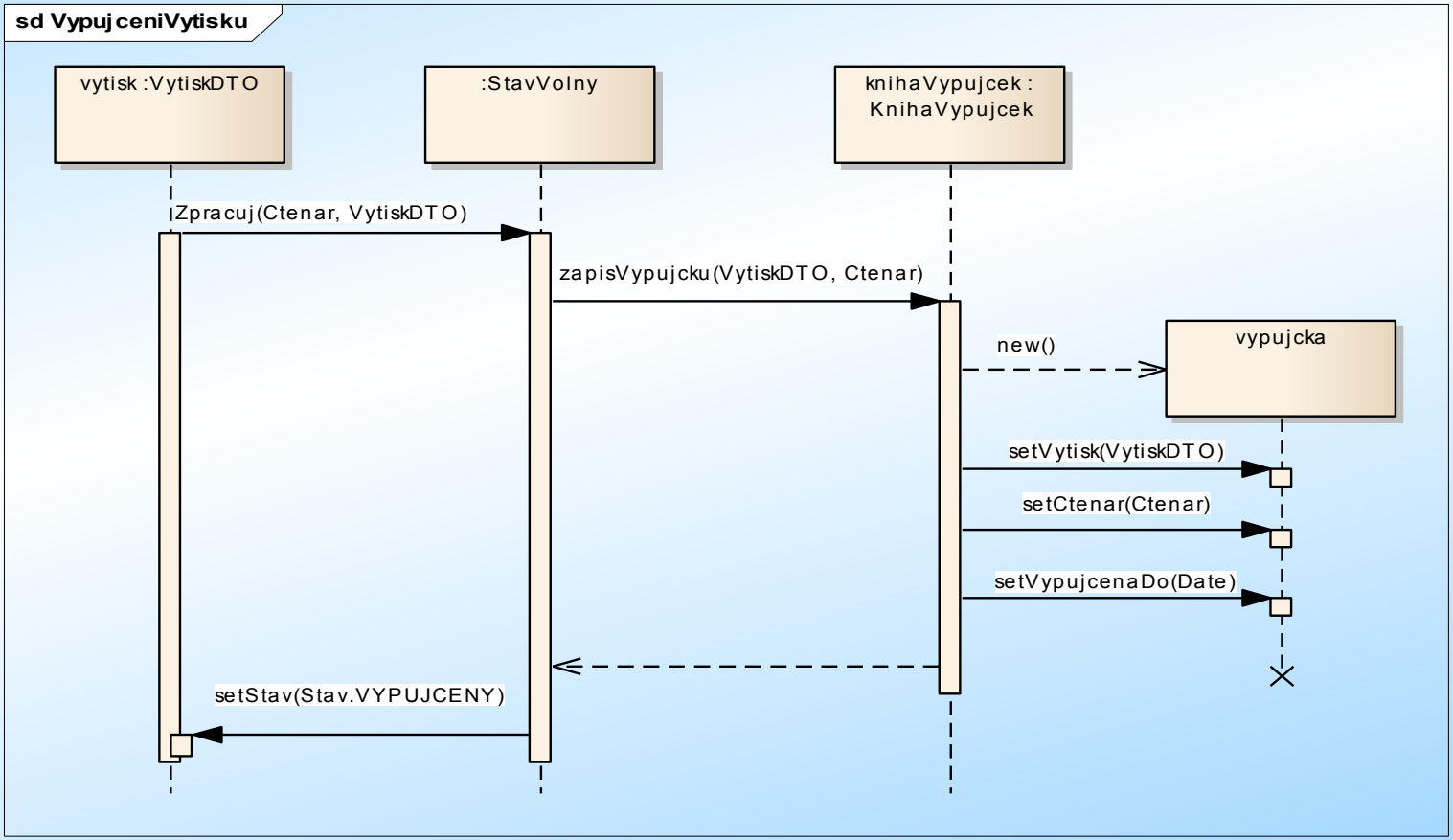
GoF vzory

- Stav (State)



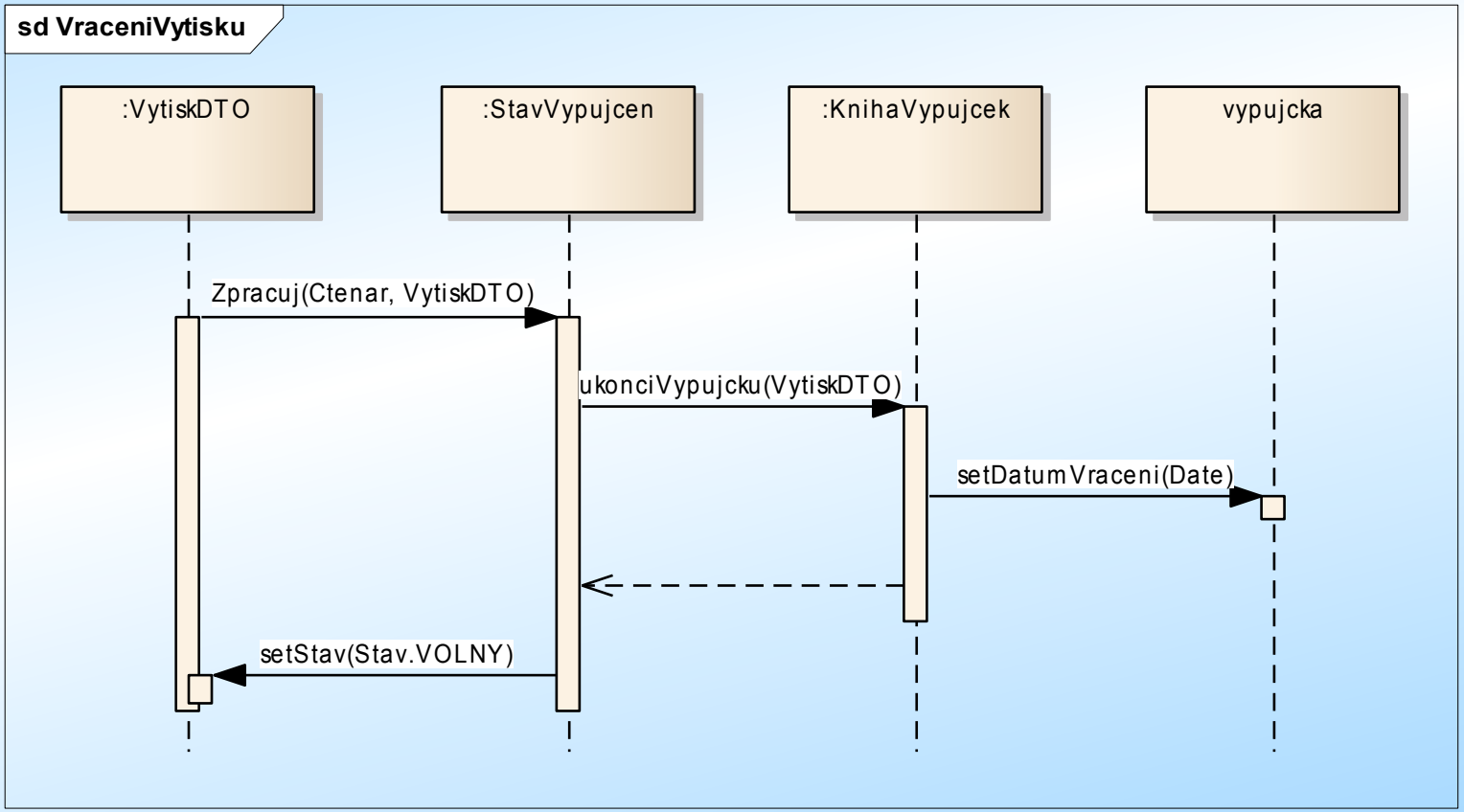
GoF vzory

- Stav (State)



GoF vzory

- Stav (State)

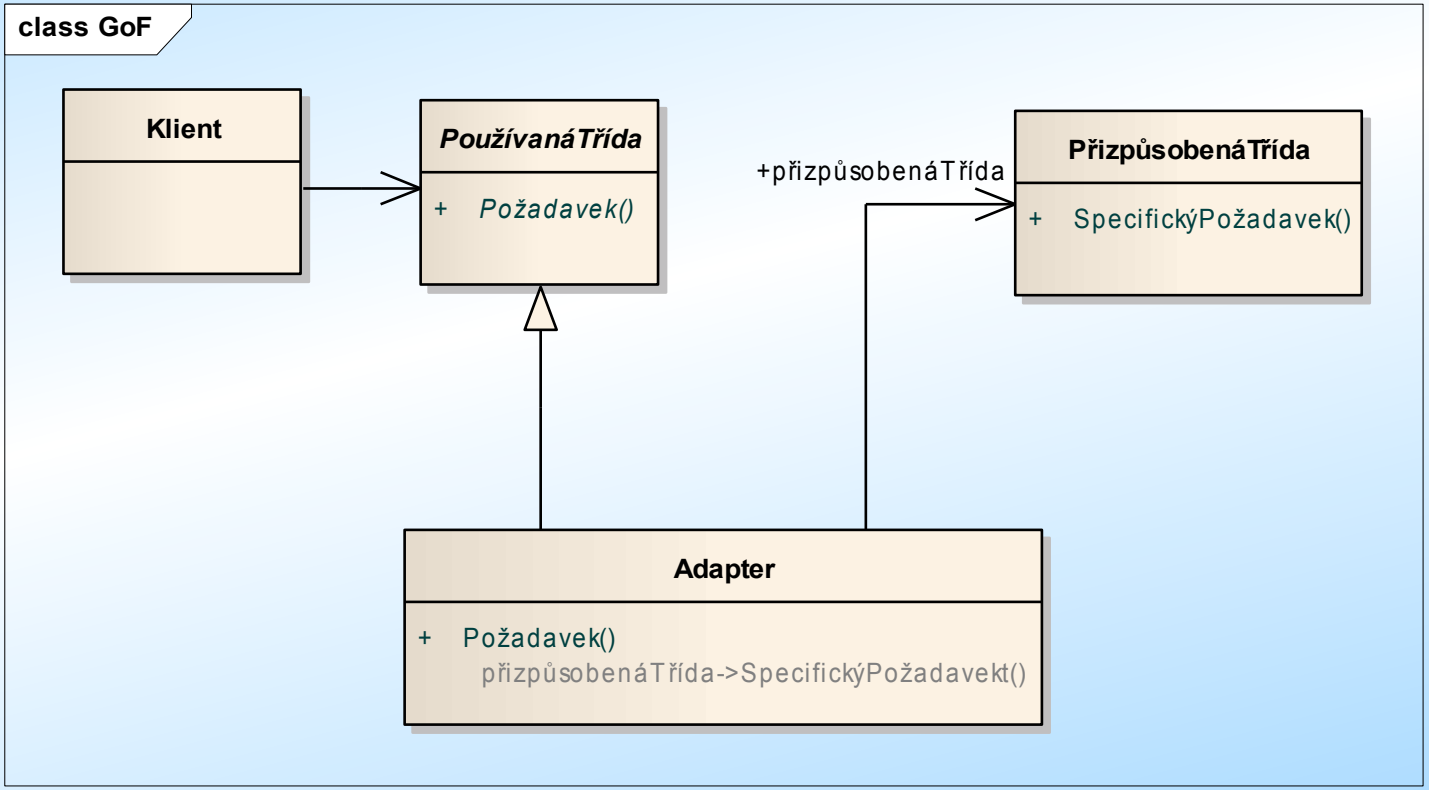


GoF vzory

- Adaptér (Adapter)
  - Konvertuje rozhraní jedné třídy na rozhraní jiné
  - Umožňuje propojit třídy s různým rozhraním

GoF vzory

- Adaptér (Adapter)



GoF vzory

Dotazy?

Děkuji za pozornost.