

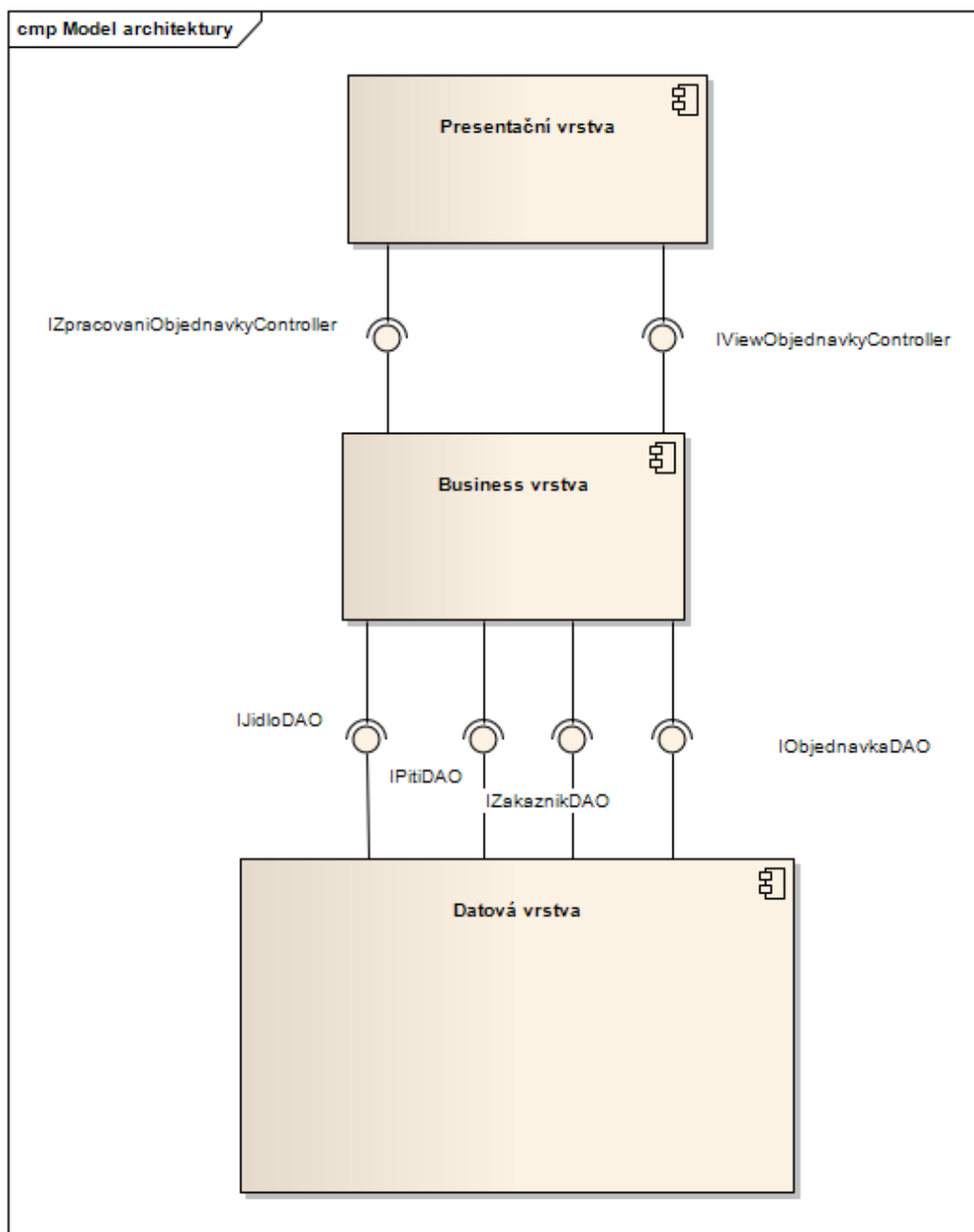
Informační systém pro Hospodu

Návrhová dokumentace

Obsah

Model architektury

Architektura aplikace je navržena jako třívrstvá. Rozhraní mezi jednotlivými vrstvami je definováno pomocí rozhraní (interface). Přesná definice těchto rozhraní umožňuje snadnou výměnu jednotlivých vrstev. Tato výměna může být požadována např. z důvodu změny uživatelského rozhraní na webovou aplikaci, výměny datové úložiště nebo pro potřeby otestování funkčností jednotlivých vrstev odděleně. Architektura také usnadňuje spolupráci více osob na projektu, kdy každý implementuje svojí část aplikace proti rozhraní bez ohledu na jeho konkrétní implementaci. Architektura aplikace je znázorněna na následujícím obrázku.



Prezentační vrstva

Prezentační vrstva obsahuje třídy a komponenty, které se starají o prezentování informací uživateli, a které od něho zpátky přebírají požadavky. Tyto požadavky transformují tak, aby odstranily závislost na použité technologii (desktopová aplikace, http request) a předávají je dále do doménového modelu.

Business vrstva

Komponenta obsahuje třídy a komponenty, které realizují požadované chování aplikace z pohledu obchodní logiky.

Datová vrstva

Komponenta obsahuje třídy a komponenty poskytující technickou podporu pro doménovou vrstvu v podobě persistentního ukládání dat apod.

Balíček "Návrhový model"

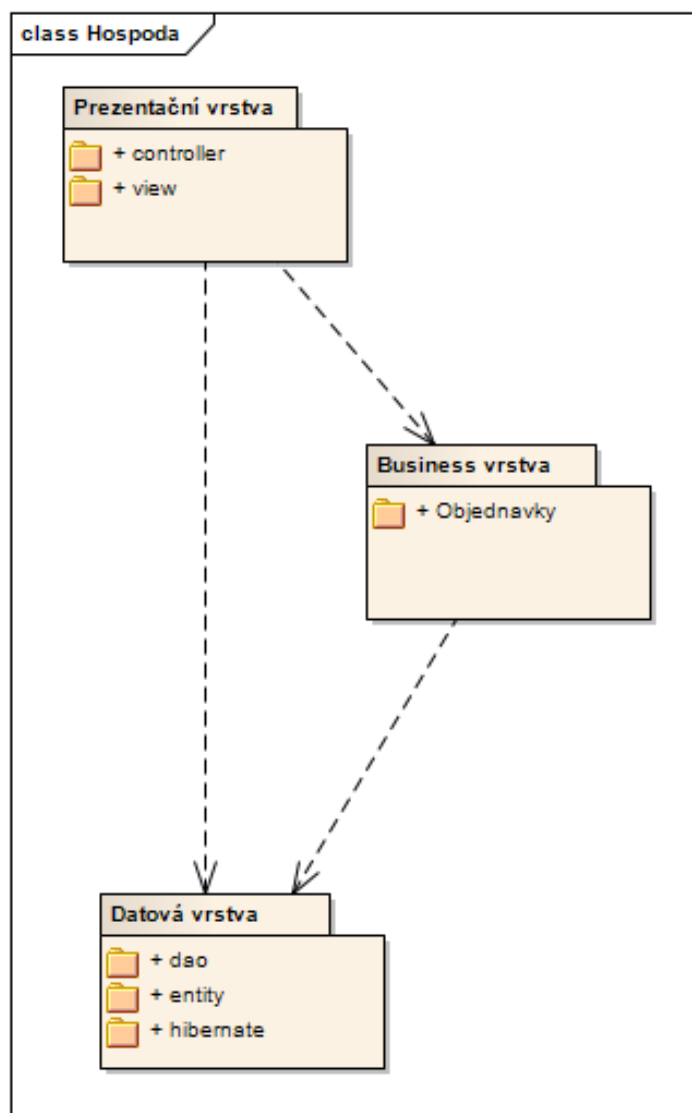
Kapitola obsahuje statický popis systému. Obsahuje detailní popis jednotlivých tříd, rozhraní a balíčků, které realizují tento systém. Kapitoly v této knize přesně odpovídají názvům java packages, které jsou použity pro seskupování zdrojových kódů tříd.

Pro implementaci celé aplikace je navrženo použití frameworku Spring, usnadňující vývoj enterprise aplikací. V tomto projektu je využito především vzoru IoC, který umožňuje provázat jednotlivé komponenty (třídy) deklarativním způsobem v konfiguraci aplikace, což umožňuje jednoduše nahradit, některé části za jiné, bez zásahu do zdrojových kódů, což může být vhodné například pro potřeby testování.



Balíček „Hospoda“

Tento balíček obsahuje tři balíčky, které reprezentují každou vrstvu systému. Schéma komunikace je uvedeno na obrázku.



Balíček „Prezentační vrstva“

Balíček obsahuje třídy prezentační vrstvy, které umožňují zobrazování stavu aplikace uživateli a předávat uživatelské požadavky do business vrstvy.

Balíček „Business vrstva“

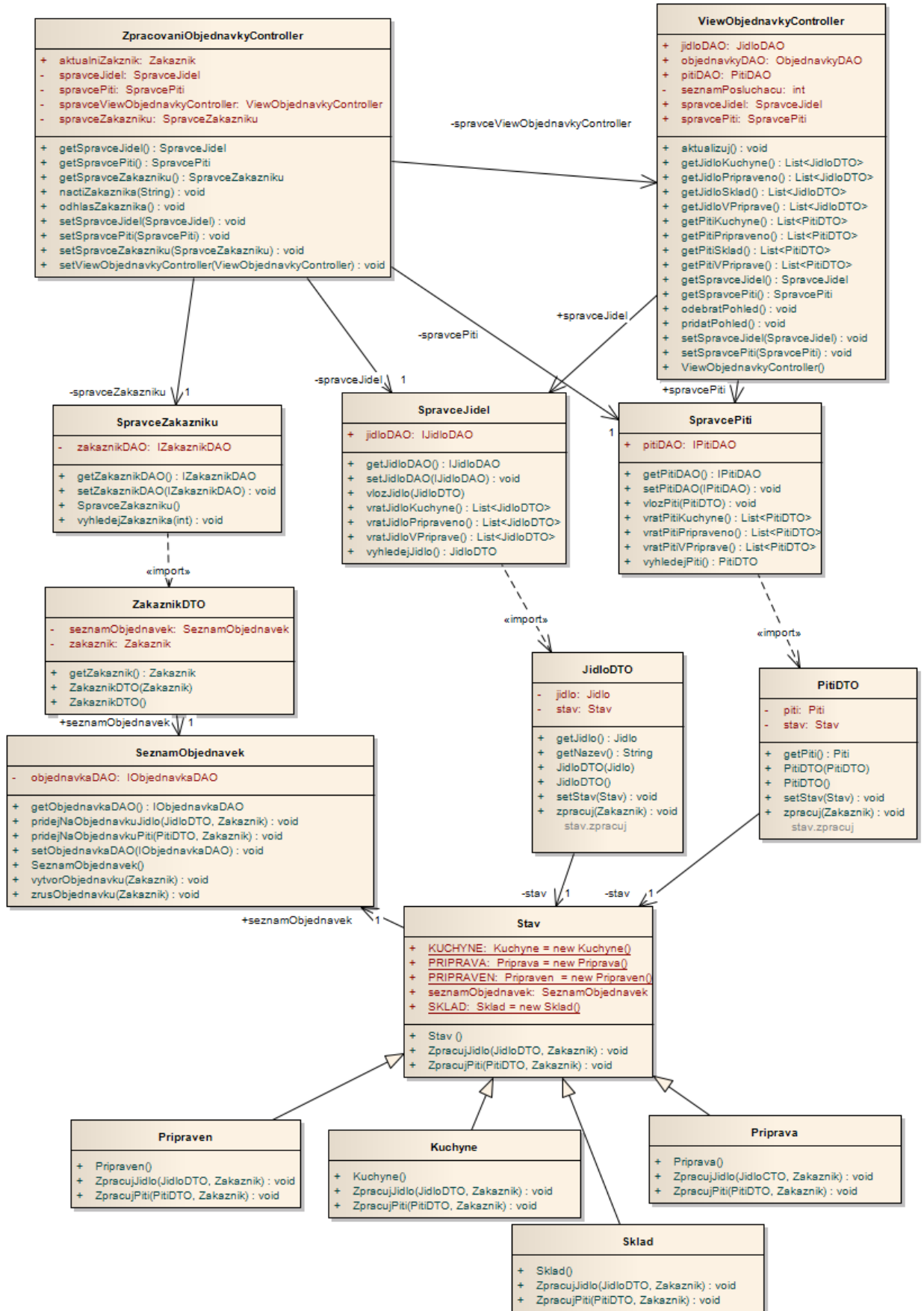
Balíček obsahuje třídy business vrstvy, které implementují vlastní chování celé aplikace. Obsahuje třídy, které realizují veškeré chování aplikace související s obchodní logikou.



Balíček „Objednavky“

Balíček obsahuje třídy, které jsem použili pro koncovou implementaci, tzn. oblast evidenci objednávek.

class Objednavky



Třída „SeznamObjednavek“

Atributy

Atributy	Datový typ	Poznámky
objednavkaDAO	IObjednavkaDAO	Rozhraní umožňuje ukládání záznamů o objednávkách.

Metody

Metody	Návratový typ	Poznámky
pridejNaObjednavkuJidlo	false	Přidá na již existující objednávky jídlo Parametry (JidloDTO, ZakaznikDTO)
setObjednavkaDAO	false	Umožňuje nastavit třídu ObjednavkaDAO Parametry (IObjednavkaDAO)
vytvorObjednavku	false	Vytvoří objednávku. Parametry (ZakaznikDTO)
zrusObjednavku	false	Zruší objednávku. Parametry (ZakaznikDTO)
odberJidloZObjednavky	false	Odebere jídlo z objednávky Parametry (ZakaznikDTO, JidloDTO)

Třída „SpravceZakazniku“

Atributy

Atributy	Datový typ	Poznámky
zakaznikDAO	IZakaznikDAO	Rozhraní umožňuje ukládání záznamů o zákaznících.

Metody

Metody	Návratový typ	Poznámky
setZakaznikDAO	False	Umožňuje nastavit třídu ZakaznikDAO Parametry (IZakaznikDAO)
vyhledejZakaznika	False	Vyhledá zákazníka na základě jeho id. Parametry (int)

Třída „SpravceJidel“

Atributy

Atributy	Datový typ	Poznámky
jidloDAO	IJidloDAO	Rozhraní umožňuje ukládání záznamů o jídlech.

Metody

Metody vrat* jsou si podobné proto je nebudu vypisovat.

Metody	Návratový typ	Poznámky
vlozJidlo	false	Přidá objednávky jídlo na seznam. Parametry (JidloDTO)
setJidloDAO	false	Umožňuje nastavit třídu JidloDAO Parametry (IJidloDAO)
vratJidloKuchyne	List<JidloDTO>	Vrátí seznam jídel, která jsou právě připravována v kuchyni.
vyhledejJidlo	JidloDTO	Na základě id jídla vrátí informace o něm.

Třída „SpravcePiti“

Viz. SpravceJidel

Třída „Stav“

Atributy

Atributy	Datový typ	Poznámky
KUCHYNE		
seznamObjednavel	SeznamObjednavek	

Metody

Metody	Návratový typ	Poznámky
zpracujJidlo	false	Parametry (JidloDTO, ZakaznikDTO)

Třída „Kuchyne“

Metody

Metody	Návratový typ	Poznámky
zpracujJidlo	false	Parametry (JidloDTO, ZakaznikDTO)

Třída „Připraven“

Viz. Kuchyně.

Třída „Priprava“

Viz. Kuchyně.

Třída „Sklad“

Viz. Kuchyně.

Třída „ViewObjednavkyController“

Atributy

Atributy	Datový typ	Poznámky
seznamPosluchacu	List<IPohled>	Seznam objektů, které chtějí být informováni o změnách v doménové vrstvě.
spravceJidel	SpravceJidel	Třída zapouzdřující logiku související z evidenci jídel.

Metody

Metody	Návratový typ	Poznámky
pridatPohled	false	Umožňuje přidat do seznamu posluchačů nový objekt. Parametry (IPohled)
setObjednavkaDAO	false	Umožňuje nastavit třídu ObjednavkaDAO Parametry (IObjednavkaDAO)
odebratPohled	false	Umožňuje odebrat objekt ze seznamu posluchačů. Parametry (IPohled)
aktualizuj	false	Všem objektům v seznamu posluchačů zašle zprávu aktualizuj.

Třída „ZakaznikDTO“

Atributy

Atributy	Datový typ	Poznámky
seznamObjednavek	SeznamObjednavek	
zakaznik	Zakaznik	Objekt, který umožňuje uložit všechny informace o zákazníkovi do databáze.

Metody

Metody	Návratový typ	Poznámky
ZakaznikDTO	false	Vytvoří novou instanci třídy.

Třída „JidloDTO“

Viz. ZakaznikDTO.

Třída „PitiDTO“

Viz. ZakaznikDTO.

Třída "ZpracovaniObjednavkyController"

Atributy

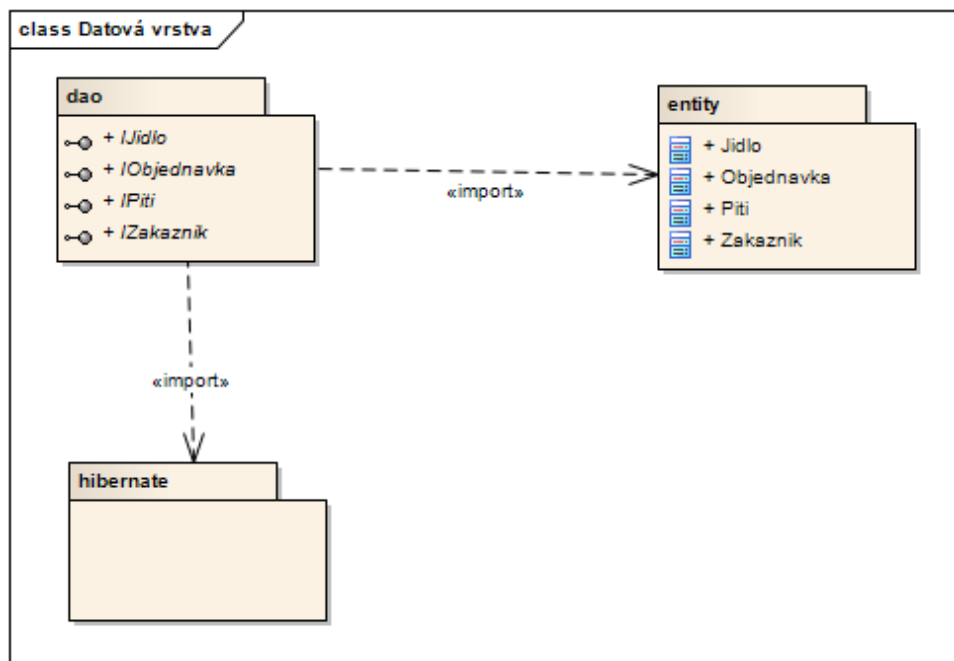
Atributy	Datový typ	Poznámky
aktualniZakaznik	Zakaznik	

Metody

Metody	Návratový typ	Poznámky
nactiZakaznika	false	Zaznamená zákazníka, ke kterému mají být následně všechny objednávky zaznamenávány. Typicky probíhá po přihlášení zákazníka do systému. Parametry (int)
odhlasZakaznika	false	Odhlásí aktuálního zákazníka ze systému.

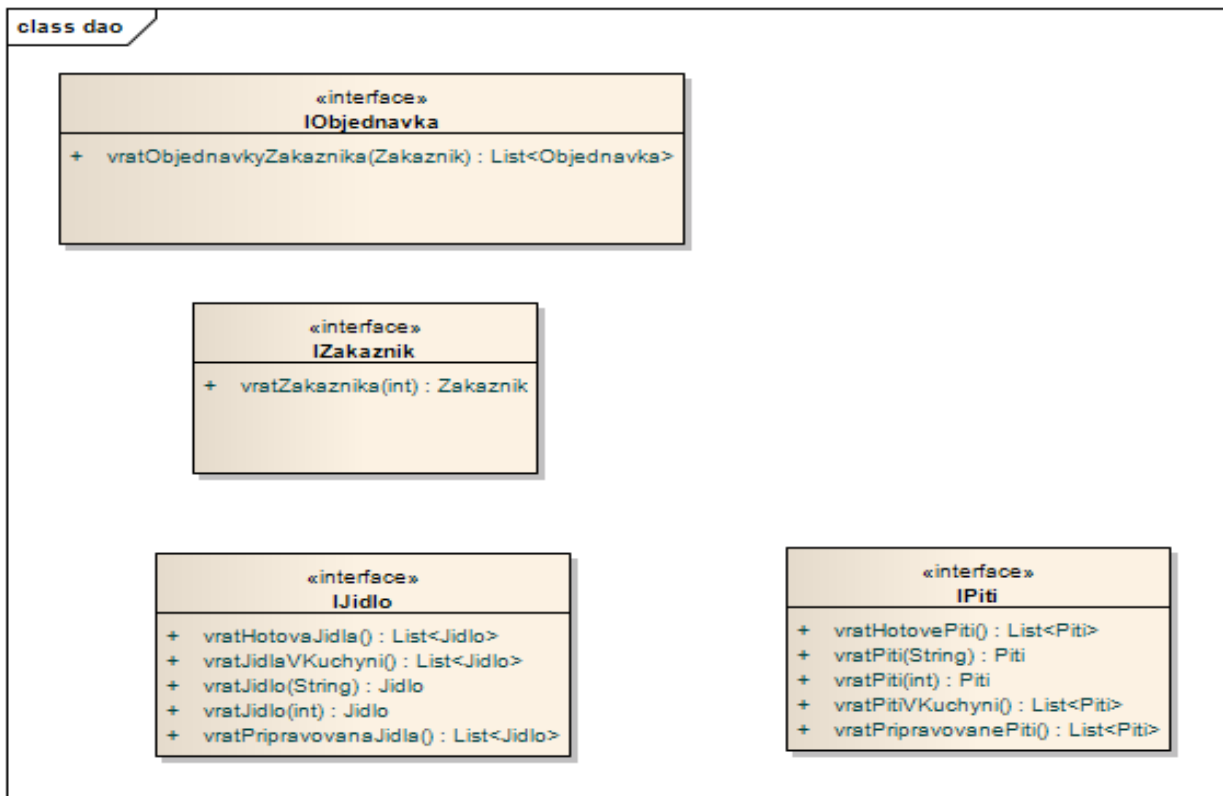
Balíček „datová vrstva“

Balíček obsahuje třídy datové vrstvy, které nabízejí technickou podporu pro třídy obchodní vrstvy. Jedná se především o persistentní ukládání informací.

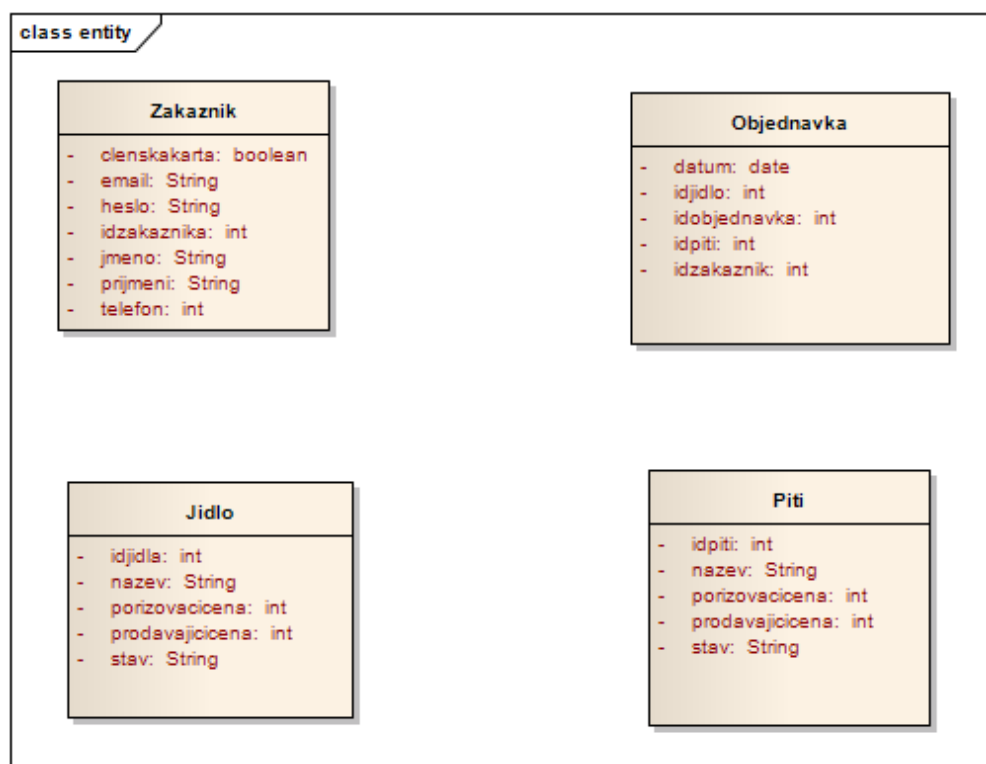


Balíček „dao“

Balíček obsahuje třídy sloužící pro ukládání a načítání persistentních objektů z/do persistentního úložiště. Obsahuje definici rozhraní persistentní vrstvy a jejich implementaci.



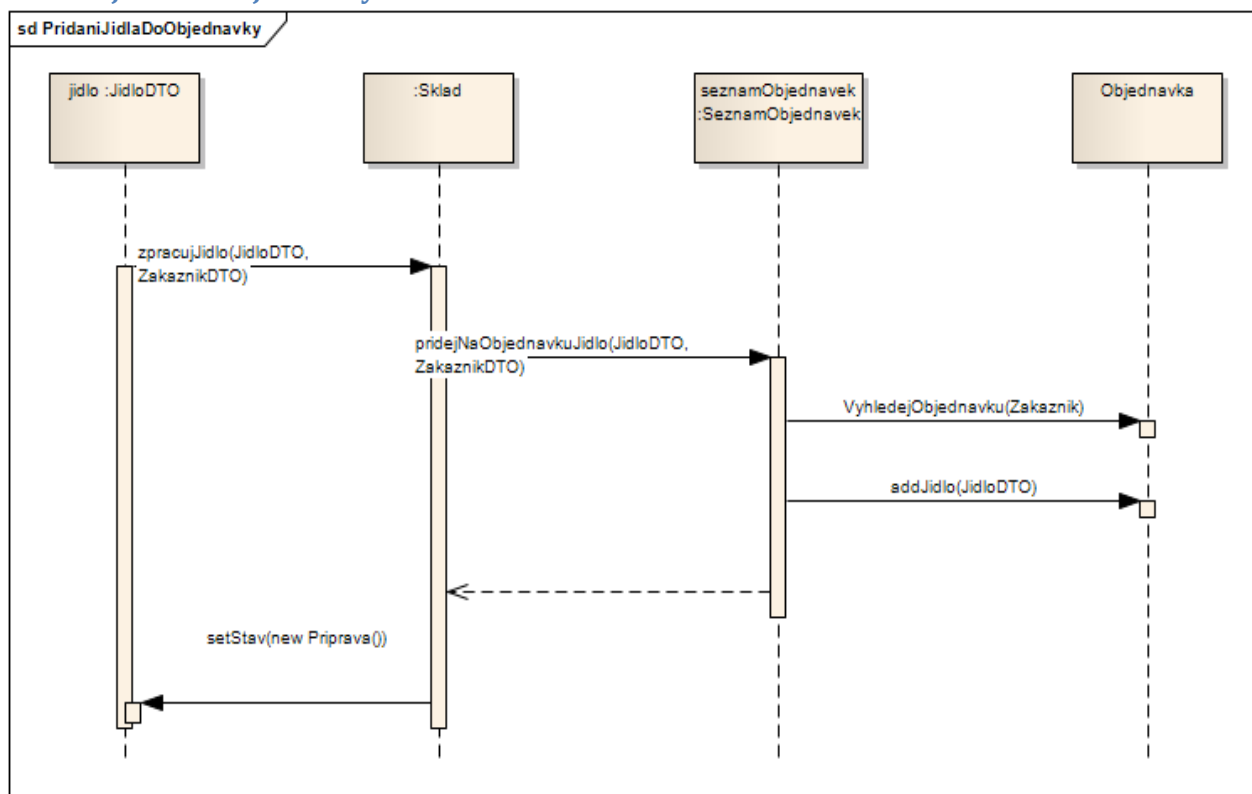
Balíček „entity“



Model komunikace

Objednávky

Přidání jídla do objednávky



Vytvoření objednávky

