

---

# Internetový obchod

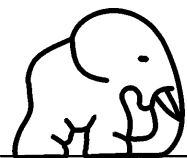
---

Semestrální projekt  
z předmětu  
Databázové a znalostní IS

---

Milka  
([www.cz-milka.net](http://www.cz-milka.net))

---



# 1. Obsah

	Strana
1. Obsah	2
2. Úvod	3
3. Relační návrh modelu	4
3.1 Entity Relationship Diagram	4
3.2 Atributy	4
3.3 SQL dotazy	6
4. Objektový návrh modelu	7
4.1 Třídy	7
4.2 Atributy	7
4.3 Množiny	8
4.4 Instance	8
4.5 Metody	13
4.6 Diagram	14
4.7 Třídy ve výstupu z Daskalu	15
4.8 Dotazy	17
4.9 Pravidla	18
5. Závěr	19
6. Seznam obrázků a tabulek	20



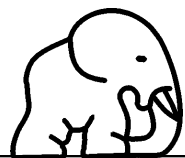
## 2. Úvod

Cílem projektu je praktické využití získaných znalostí z oboru projektování informačních systémů. Předmětem práce je návrh objektového a relačního modelu fiktivního internetového obchodu. Relační model bude navržen s využitím nástroje MetaEdit+, k návrhu objektového modelu bude použit Daskalos 1.02b.

Předložený model bude popisovat obchod, který zákazníkům prodává počítačové komponenty výhradně přes Internet. V modelu budou znázorněny pouze základní vztahy mezi zúčastněnými subjekty. Předpokladem je, že dodavatel i odběratel se nejprve musí zaregistrovat do interního systému firmy. Externí dodavatel po registraci získá možnost nabídnout své zboží internetovému obchodu. Odběratel má pak možnost zboží nakupovat. Zákazníky internetového obchodu mohou být fyzické i právnické osoby.

Objednávka odběratele se vyřizuje zjištěním dostupnosti zboží na skladě. Je-li zboží nedostupné, bude objednáno, dodáno dodavatelem a uloženo na sklad. Je-li dostupné, je vyskladněno a předáno společně s vystavenou fakturou odběrateli. Faktury přijaté od dodavatelů nebudou v modelu uvažovány. Odběratelé své závazky hradí výhradně pomocí bankovních příkazů na jeden z účtů, kterým internetový obchod disponuje.

Model nebude uvažován v úplnosti, jelikož věrné zachycení reality není hlavním cílem této práce. V projektu tedy dojde k simplifikaci, jelikož výsledný model by byly příliš komplikovaný. Internetový obchod například neumožňuje stornování objednávky, reklamace vadného zboží nebo vracení peněz odběratelům.

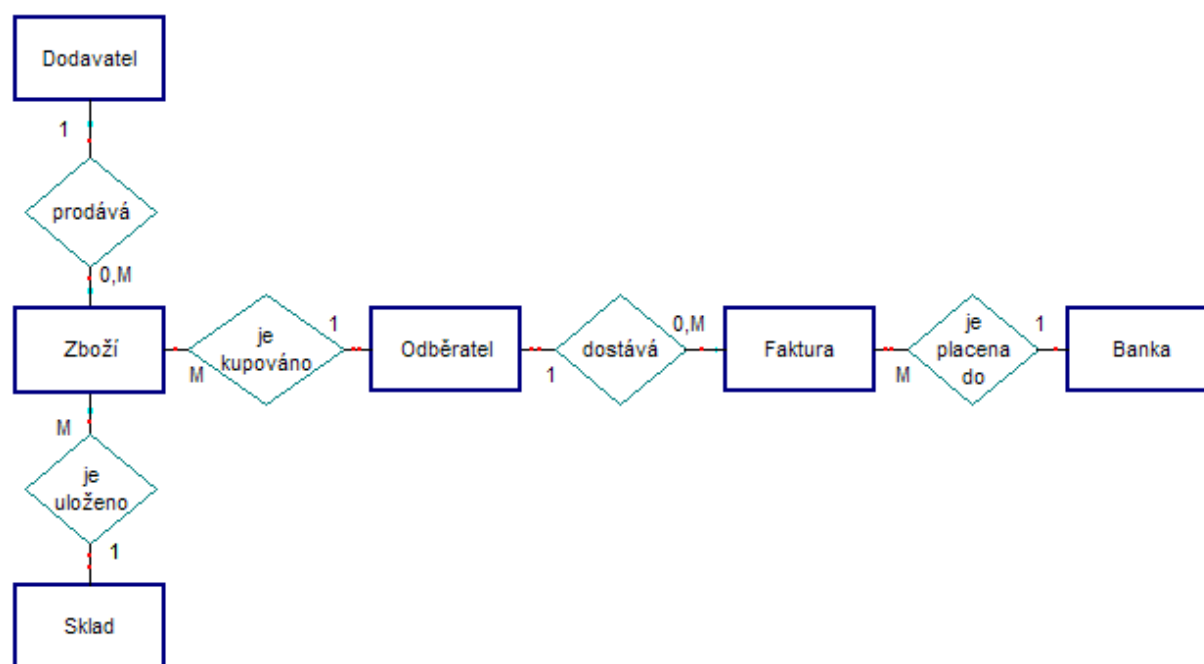


## 3. Relační návrh modelu

### 3.1 Entity Relationship Diagram

Relační model byl navržen s využitím case nástroje MetaEdit+. K navržení tohoto modelu byl využit Entity Relationship Diagram (ERD), tedy diagram vztahů a entit. Tento diagram modeluje vnitřní datovou stránku informačního systému a sestává z entit a vazeb mezi nimi. Pomocí ERD je možné také popsat atributy přiřazené entitám nebo vztahům, které se většinou uvádějí v datovém slovníku.

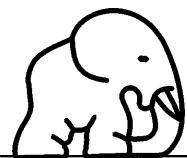
Při návrhu relačního modelu, na rozdíl od objektového, bylo předpokládáno, že společnost má pouze jedny skladovací prostory a účet pouze u jedné bankovní instituce.



Obr. 1 – Entity Relationship Diagram.

### 3.2 Atributy

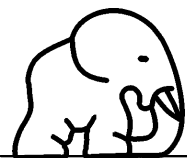
Součástí návrhu Entity Relationship Diagramu je také přiřazení atributů jednotlivým entitám a vztahům. Jelikož atributy vztahů by v tomto případě nepřinesly žádné vylepšení, budou přiřazeny atributy pouze entitám.



Entita	Atributy	Typ	Omezení
Dodavatel	ID dodavatele ID zboží Jméno/název IČ DIČ Adresa Email Telefon Číslo účtu	Integer Integer Varchar (50) Varchar (10) Varchar (15) Varchar (100) Varchar (50) Varchar (15) Varchar (20)	Primary Key, Unique, Not Null Foreign Key, Not Null Not Null - - Not Null Not Null - Not Null
Odběratel	ID odběratele ID zboží ID faktury Jméno/název Rok narození IČ DIČ Adresa Email Telefon	Integer Integer Integer Varchar (50) Float (4) Varchar (10) Varchar (15) Varchar (100) Varchar (50) Varchar (15)	Primary Key, Unique, Not Null Foreign Key, Not Null Foreign Key, Not Null Not Null - - - Not Null Not Null -
Faktura	ID faktury ID zboží ID banky Cena Datum zdanitelného plnění Datum splatnosti	Integer Integer Integer Float (10,2) Datetime Datetime	Primary Key, Unique, Not Null Foreign Key, Not Null Foreign Key, Not Null Not Null Not Null Not Null
Zboží	ID zboží ID skladu Název zboží Výrobce Cena	Integer Integer Varchar (50) Varchar (20) Float (10,2)	Primary Key, Unique, Not Null Foreign Key, Not Null Not Null Not Null Not Null
Sklad	ID skladu Typ	Integer Varchar (20)	Primary Key, Unique, Not Null -
Banka	ID banky Název Adresa Číslo účtu Kód banky	Integer Varchar (20) Varchar (100) Varchar (15) Varchar (5)	Primary Key, Unique, Not Null Not Null Not Null Not Null Not Null

Tab. 1 – Atributy entit.

Jak z výše uvedené tabulky vyplývá, propojenost tabulek v relačním modelu je řešena pomocí cizího klíče. U relačního modelu není možné znárodnit dědění, které bude zobrazeno v modelu objektovém.



## 3.3 SQL dotazy

Jako příklad si uveďme několik SQL dotazů. Jejich objektová obdoba bude uvedena v kapitole 4.8.

- » Výběr všech odběratelů, kteří se jmenují Jan Koláček:

```
SELECT * FROM Odberatele WHERE jmeno_nazev LIKE 'Kolacek Jan';
```

- » Výběr zboží, které je levnější než 2000 Kč:

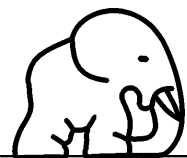
```
SELECT * FROM Zbozi_vse WHERE cena < 2000;
```

- » Zjištění, do jakých bank byly placeny faktury:

```
SELECT * FROM Faktury LEFT JOIN Banky ON  
Faktury.id_banky = Banky.id_banky GROUP BY Banky.id_banky;
```

- » Zjištění výrobců zboží, které stojí méně, než 5000 Kč:

```
SELECT * FROM Zbozi_vse WHERE cena < 5000 GROUP BY vyrobce;
```



## 4. Objektový návrh modelu

Objektový model bude navržen pomocí programu Daskalos.

V prvním kroku je nutné vyplnit údaje o novém projektu, tedy název, autora a stručný popis, jelikož bez příslušné dokumentace je jakýkoli projekt nepoužitelný.

### 4.1 Třídy

Model bude uchovávat informace o dodavatelích, odběratelích, zboží a jeho uložení, fakturacích a platbách.

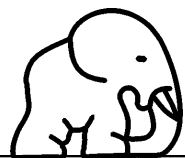
V modelu budou použity tyto třídy:

- » Registrovaný uživatel – třída, která bude předchůdcem třídy Dodavatel a třídy Odběratel.
- » Dodavatel – třída, která uchovává informace o subjektech, které dodávají do internetového obchodu zboží.
- » Odběratel – tato třída uchovává informace o zákaznících.
- » Faktura – jedná se o třídu, která uchovává informace o fakturách vystavených na prodané zboží.
- » Zboží – třída, která uchovává informace o zboží, jeho výrobci a ceně.
- » Sklad – tato třída uchovává informace o aktuálním stavu zásob.
- » Banka – třída uchovávající informace o platbách, které provedli odběratelé.

### 4.2 Atributy

Všem třídám je nutné přiřadit atributy:

Třída	Atributy	Typ
Registrovaný uživatel	Jméno/název IČ DIČ Adresa Email Telefon	String String String String String String
Dodavatel	Číslo účtu	String
Odběratel	Rok narození	Number
Faktura	ID faktury Datum zdanitelného plnění Datum splatnosti	Number Date Date



Zboží	ID zboží Název zboží Výrobce Cena	Number String String Number
Sklad	ID skladu Typ	Number String
Banka	Název Adresa Číslo účtu Kód banky	String String String String

Tab. 2 – Atributy tříd.

Je nutné zdůraznit, že uváděné atributy id neodpovídají relačnímu modelu. Například v případě faktury představuje atribut „ID faktury“ jedinečné číslo faktury, respektive variabilní symbol, nebo v případě zboží představuje „ID zboží“ jedinečné katalogové číslo.

Jakmile máme definovány všechny potřebné třídy, můžeme vytvořit množiny a konkrétní instance.

## 4.3 Množiny

V dalším kroku modelujeme množiny – můžeme buď ručně pomocí tlačítka „new object“ nebo, jako v našem případě, v záložce „script“.

```
Uzivatele:= Set new.  
Dodavatele:= Set new.  
Odberatele:= Set new.  
Banky:= Set new.  
Faktury:= Set new.  
Zbozi_vse:= Set new.  
Sklady:= Set new.
```

## 4.4 Instance

Model doplníme o data, tedy vytvoříme konkrétní objekty – instance. V našem případě využijeme „příkazového řádku“ v záložce „script“.

Plné znění skriptu:

```
B1:= Banka new.  
B1 nazev: 'Ceska Sporitelna'.  
B1 adresa: 'P8, Lodzska 598'.  
B1 cislo_uctu: '1454125412'.  
B1 kod_banky: '0800'.  
Banky add: B1.
```





B2:= Banka new.  
B2 nazev: 'CSOB'.  
B2 adresa: 'P8, Ke Stirce 50'.  
B2 cislo\_uctu: '5879856897'.  
B2 kod\_banky: '0300'.  
Banky add: B2.

B3:= Banka new.  
B3 nazev: 'E-Banka'.  
B3 adresa: 'P8, Kyselova 1658'.  
B3 cislo\_uctu: '8465875654'.  
B3 kod\_banky: '2400'.  
Banky add: B3.

B4:= Banka new.  
B4 nazev: 'MBank'.  
B4 adresa: 'P2, Vinohradska 938/37'.  
B4 cislo\_uctu: '36936569365'.  
B4 kod\_banky: '6210'.  
Banky add: B4.

S1:= Sklad new.  
S1 id\_skladu: '1' asNumber.  
S1 typ: 'HW - objemne'.  
Sklady add: S1.

S2:= Sklad new.  
S2 id\_skladu: '2' asNumber.  
S2 typ: 'HW - krehke'.  
Sklady add: S2.

S3:= Sklad new.  
S3 id\_skladu: '3' asNumber.  
S3 typ: 'HW - ostatni'.  
Sklady add: S3.

S4:= Sklad new.  
S4 id\_skladu: '4' asNumber.  
S4 typ: 'SW'.  
Sklady add: S4.

Z1:= Zbozi new.  
Z1 id\_zbozi: '1' asNumber.  
Z1 nazev\_zbozi: 'Herni konzole XBOX 360 Pro'.  
Z1 vyrobce: 'Logitech'.  
Z1 cena: '8318' asNumber.  
Z1 je\_ulozeno: S3.  
Zbozi\_vse add: Z1.

Z2:= Zbozi new.  
Z2 id\_zbozi: '2' asNumber.  
Z2 nazev\_zbozi: 'Zakladni deska MB Asus M2A-MX'.  
Z2 vyrobce: 'Asus'.  
Z2 cena: '998' asNumber.  
Z2 je\_ulozeno: S2.  
Zbozi\_vse add: Z2.



```
Z3:= Zbozi new.  
Z3 id_zbozi: '3' asNumber.  
Z3 nazev_zbozi: 'Procesor Intel Pentium Dual Core E2160'.  
Z3 vyrobce: 'Intel'.  
Z3 cena: '1416' asNumber.  
Z3 je_ulozeno: S2.  
Zbozi_vse add: Z3.
```

```
Z4:= Zbozi new.  
Z4 id_zbozi: '4' asNumber.  
Z4 nazev_zbozi: 'Procesor AMD Athlon A64 X2 6000+'.  
Z4 vyrobce: 'AMD'.  
Z4 cena: '3299' asNumber.  
Z4 je_ulozeno: S2.  
Zbozi_vse add: Z4.
```

```
Z5:= Zbozi new.  
Z5 id_zbozi: '5' asNumber.  
Z5 nazev_zbozi: 'Monitor 19'' LG LCD L194WT-SF 5ms'.  
Z5 vyrobce: 'Lg'.  
Z5 cena: '4375' asNumber.  
Z5 je_ulozeno: S1.  
Zbozi_vse add: Z5.
```

```
F1:= Faktura new.  
F1 id_faktury: '1' asNumber.  
F1 datum_zdanitelneho_plneni: '4-11-2008' asDate.  
F1 datum_splatnosti: '4-21-2008' asDate.  
F1 obsahuje add: Z1.  
F1 obsahuje add: Z5.  
F1 je_placena_do add: B1.  
Faktury add: F1.
```

```
F2:= Faktura new.  
F2 id_faktury: '2' asNumber.  
F2 datum_zdanitelneho_plneni: '4-13-2008' asDate.  
F2 datum_splatnosti: '4-23-2008' asDate.  
F2 obsahuje add: Z2.  
F2 je_placena_do add: B1.  
Faktury add: F2.
```

```
F3:= Faktura new.  
F3 id_faktury: '3' asNumber.  
F3 datum_zdanitelneho_plneni: '4-15-2008' asDate.  
F3 datum_splatnosti: '4-25-2008' asDate.  
F3 obsahuje add: Z4.  
F3 je_placena_do add: B2.  
Faktury add: F3.
```

```
D1:= Dodavatel new.  
D1 jmeno_nazev: 'Mironet'.  
D1 ic: '25728946'.  
D1 dic: 'CZ25728946'.  
D1 adresa: 'P7, V haji 10'.  
D1 email: 'prodej@mironet.cz'.  
D1 telefon: '234700800'.  
D1 cislo_uctu: '1255332036/2400'.  
D1 prodava add: Z1.  
D1 prodava add: Z3.  
D1 prodava add: Z5.  
Dodavatele add: D1.  
Uzivatele add: D1.
```



D2:= Dodavatel new.  
D2 jmeno\_nazev: 'Alzasoft'.  
D2 ic: '270082440'.  
D2 dic: 'CZ270082440'.  
D2 adresa: 'P7, Jateční 3a'.  
D2 email: 'objednavka@alzasoft.cz'.  
D2 telefon: '234111222'.  
D2 cislo\_uctu: '188505042/0300'.  
D2 prodava add: Z2.  
D2 prodava add: Z4.  
Dodavatele add: D2.  
Uzivatele add: D2.

D3:= Dodavatel new.  
D3 jmeno\_nazev: 'CzechComputer'.  
D3 ic: '25655701'.  
D3 dic: 'CZ25655701'.  
D3 adresa: 'P4, Hviezdoslavova 666'.  
D3 email: 'obchod@czc.cz'.  
D3 telefon: '242410000'.  
D3 cislo\_uctu: '157415424/0300'.  
Dodavatele add: D3.  
Uzivatele add: D3.

D4:= Dodavatel new.  
D4 jmeno\_nazev: 'Kasa.cz'.  
D4 ic: '26772884'.  
D4 dic: 'CZ26772884'.  
D4 adresa: 'P9, Kostelecká 879'.  
D4 email: 'info@kasa.cz'.  
D4 telefon: '841800800'.  
D4 cislo\_uctu: '157415424/0300'.  
Dodavatele add: D4.  
Uzivatele add: D4.

O1:= Odberatel new.  
O1 jmeno\_nazev: 'Kolacek Jan'.  
O1 rok\_narozeni: '1983' asNumber.  
O1 ic: ''.  
O1 dic: ''.  
O1 adresa: 'P4, Pekarska 14'.  
O1 email: 'kolacek@seznam.cz'.  
O1 telefon: '603327421'.  
O1 kupuje add: Z1.  
O1 kupuje add: Z5.  
O1 dostava add: F1.  
Odberatele add: O1.  
Uzivatele add: O1.

O2:= Odberatel new.  
O2 jmeno\_nazev: 'Vytvarnik.cz, s.r.o.'.  
O2 rok\_narozeni: '' asNumber.  
O2 ic: '12455643'.  
O2 dic: 'CZ12455643'.  
O2 adresa: 'P1, Václavské náměstí 1'.  
O2 email: 'info@vytvarnik.cz'.  
O2 telefon: '211485245'.  
O2 kupuje add: Z2.  
O2 dostava add: F2.  
Odberatele add: O2.  
Uzivatele add: O2.



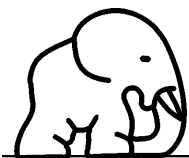
O3:= Odberatel new.  
O3 jmeno\_nazev: 'Truhlik Pavel'.  
O3 rok\_narozeni: '1973' asNumber.  
O3 ic: ''.  
O3 dic: ''.  
O3 adresa: 'P3, Kytickova 2905'.  
O3 email: 'truhlik@gmail.com'.  
O3 telefon: '602345345'.  
O3 kupuje add: Z4.  
O3 dostava add: F3.  
Odberatele add: O3.  
Uzivatele add: O3.

O4:= Odberatel new.  
O4 jmeno\_nazev: 'Kotatko Martin'.  
O4 rok\_narozeni: '1979' asNumber.  
O4 ic: ''.  
O4 dic: ''.  
O4 adresa: 'Melnik, Kockovaci 3490'.  
O4 email: 'kotatko.martin@seznam.cz'.  
O4 telefon: '7758454587'.  
Odberatele add: O4.  
Uzivatele add: O4.

O5:= Odberatel new.  
O5 jmeno\_nazev: 'Cesky Kutil.cz'.  
O5 rok\_narozeni: '' asNumber.  
O5 ic: '27594181'.  
O5 dic: 'CZ27594181'.  
O5 adresa: 'P4, Pecharova 498/5'.  
O5 email: 'info@ceskykutil.cz'.  
O5 telefon: '222154856'.  
Odberatele add: O5.  
Uzivatele add: O5.

O6:= Odberatel new.  
O6 jmeno\_nazev: 'Kaplicka Karel'.  
O6 rok\_narozeni: '1985' asNumber.  
O6 ic: ''.  
O6 dic: ''.  
O6 adresa: 'P6, Kostelni 2'.  
O6 email: 'xkapk100@studenti.pef.czu.cz'.  
O6 telefon: '605012468'.  
Odberatele add: O6.  
Uzivatele add: O6.

O7:= Odberatel new.  
O7 jmeno\_nazev: 'Stihacka Alois'.  
O7 rok\_narozeni: '1957' asNumber.  
O7 ic: ''.  
O7 dic: ''.  
O7 adresa: 'P10, Vzdušné obrany 5894'.  
O7 email: 'stihacka.a@bata.cz'.  
O7 telefon: '773145368'.  
Odberatele add: O7.  
Uzivatele add: O7.



## 4.5 Metody

Některé informace nemusíme do instanci zadávat přímo jako atributy, ale je možné tyto údaje vytvořit pomocí metody.

V případě třídy `Registrovani_uzivatel` vytvoříme metodu, která bude na základě hodnoty IČA zjišťovat, zda je dotyčný uživatel fyzická nebo právnická osoba. V případě fyzické osoby bude hodnota `true`. Zápis bude tento:

```
fyzicka_osoba
    ^self ic = ''
```

U odběratelů je vhodné evidovat věk, který můžeme využít pro další metody. Zjistíme ho tak, že od současného roku odečteme rok narození:

```
vek
    ^Date today year - rok_narozeni
```

Pokud chceme studentům do 26 let poskytovat slevu na prodávané zboží, můžeme vypočítaný věk využít pro další metodu:

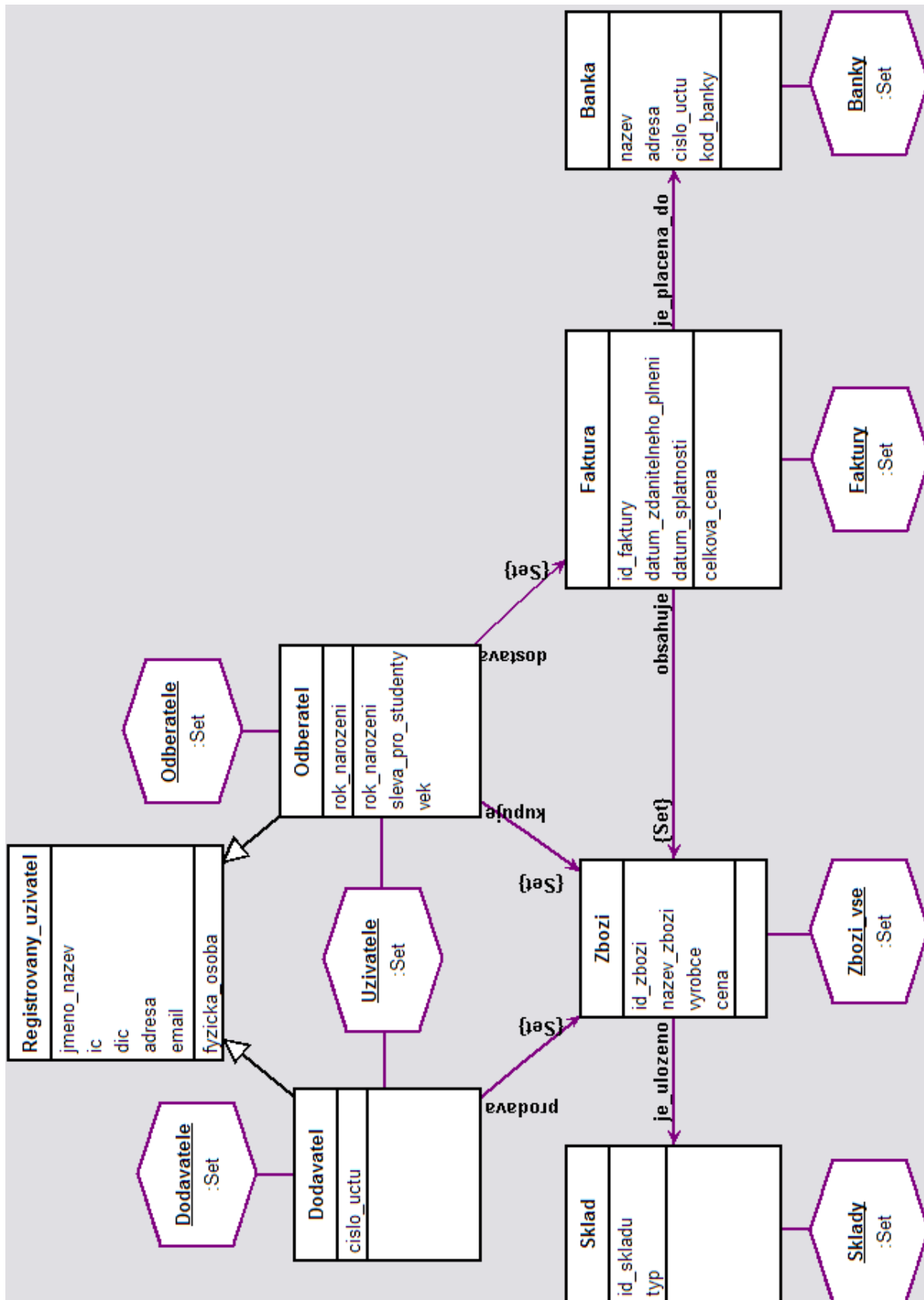
```
sleva_pro_studenty
    ^self vek < 26
```

Poslední metodou, která byla použita, je výpočet celkové ceny zboží uvedeného na faktuře:

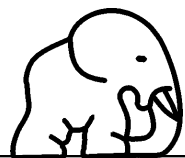
```
celkova_cena
    ^self obsahuje sum: [:x | x cena]
```



## 4.6 Diagram



Obr. 2 – Objektový model.



## 4.7 Třídy ve výstupu z Daskalu

Výsledné třídy budou vypadat, dle výstupu z programu Daskalos, takto:

Registrovany_uzivatel
<i>instance variables</i> adresa :String dic :String email :String ic :String jmeno_nazev :String telefon :String
<i>methods</i> adresa adresa: dic dic: email email: fyzicka_osoba ic ic: initialize jmeno_nazev jmeno_nazev: telefon telefon:

Registrovaný uživatel – třída, která bude předchůdcem třídy Dodavatel a třídy Odběratel.

**code of non-accessing methods:**

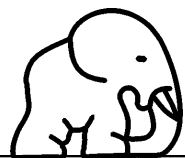
- initialize  
"generated by Daskalos"  
  
super initialize.  
jmeno\_nazev := nil.  
ic := nil.  
dic := nil.  
adresa := nil.  
email := nil.  
telefon := nil.
- fyzicka\_osoba  
^self ic = ''

Odberatel
<i>instance variables</i> dostava :Set kupuje :Set rok_narozeni :Number
<i>methods</i> dostava initialize kupuje rok_narozeni rok_narozeni: sleva_pro_studenty vek

Odběratel – tato třída uchovává informace o zákaznících.

**code of non-accessing methods:**

- initialize  
"generated by Daskalos"  
  
super initialize.  
kupuje := Set new.  
dostava := Set new.  
rok\_narozeni := nil.
- rok\_narozeni  
"generated by Daskalos"  
^rok\_narozeni
- vek  
^Date today year - rok\_narozeni
- sleva\_pro\_studenty  
^self vek < 26



Dodavatel
<i>instance variables</i> cislo_uctu :String prodava :Set
<i>methods</i> cislo_uctu cislo_uctu: initialize prodava

Dodavatel - třída, která uchovává informace o subjektech, které dodávají do internetového obchodu zboží.

**code of non-accessing methods:**

- initialize  
"generated by Daskalos"  
  
super initialize.  
cislo\_uctu := nil.  
prodava := Set new.

Faktura
<i>instance variables</i> datum_splatnosti :Date datum_zdanitelného_plnění :Date id_faktury :Number je_placena_do :Set obsahuje :Set
<i>methods</i> celkova_cena datum_splatnosti datum_splatnosti: datum_zdanitelného_plnění datum_zdanitelného_plnění: id_faktury id_faktury: initialize je_placena_do obsahuje

Faktura – jedná se o třídu, která uchovává informace o fakturách vystavených na prodané zboží.

**code of non-accessing methods:**

- initialize  
"generated by Daskalos"  
  
super initialize.  
id\_faktury := nil.  
datum\_zdanitelného\_plnění := nil.  
datum\_splatnosti := nil.  
obsahuje := Set new.  
je\_placena\_do := Set new.
- celkova\_cena  
^self obsahuje sum: [:x | x cena]

Zbozi
<i>instance variables</i> cena :Number id_zbozi :Number je_ulozeno :Object nazev_zbozi :String vyrobce :String
<i>methods</i> cena cena: id_zbozi id_zbozi: initialize je_ulozeno je_ulozeno: nazev_zbozi nazev_zbozi: vyrobce vyrobce:

Zboží – třída, která uchovává informace o zboží, jeho výrobci a ceně.

**code of non-accessing methods:**

- initialize  
"generated by Daskalos"  
  
super initialize.  
id\_zbozi := nil.  
nazev\_zbozi := nil.  
vyrobce := nil.  
cena := nil.  
je\_ulozeno := nil.





Sklad
<i>instance variables</i> id_skladu :Number typ :String
<i>methods</i> id_skladu id_skladu: initialize typ typ:

Sklad – tato třída uchovává informace o aktuálním stavu zásob.

**code of non-accessing methods:**

- initialize  
"generated by Daskalos"  
  
super initialize.  
id\_skladu := nil.  
typ := nil.

Banka
<i>instance variables</i> adresa :String cislo_uctu :String kod_banky :String nazev :String
<i>methods</i> adresa adresa: cislo_uctu cislo_uctu: initialize kod_banky kod_banky: nazev nazev:

Banka – třída uchovávající informace o platbách, které provedli odběratelé.

**code of non-accessing methods:**

- initialize  
generated by Daskalos"  
  
super initialize.  
nazev := nil.  
adresa := nil.  
cislo\_uctu := nil.  
kod\_banky := nil.

## 4.8 Dotazy

S daty je možné provádět různé operace, což vede k ověření a vyzkoušení předloženého modelu. Jako příklad operací s kolekcemi objektů uveďme:

- » Výběr všech odběratelů, kteří se jmenují Jan Koláček:

```
(Odberatele select: [:x | x jmeno_nazev = 'Kolacek Jan']).
```

- » Výběr zboží, které je levnější než 2000 Kč:

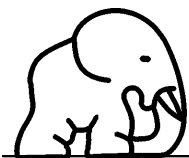
```
(Zbozi_vse select: [:x | x cena < 2000]).
```

- » Zjištění, do jakých bank byly placeny faktury:

```
(Faktury collect: [:x | x je_placena_do]).
```

- » Zjištění výrobců zboží, které stojí méně, než 5000 Kč:

```
(Zbozi_vse select: [:x | x cena < 5000]) collect: [:x | x vyrobce].
```



## 4.9 Pravidla

Pravidla jsou dotazy, které vrací true nebo false. V SQL to mohou být dotazy, které v případě true vrátí plnou množinu a v případě false množinu prázdnou. Příkladem pravidel v Daskalu jsou:

- » Nárok na studentskou slevu mají všichni narození po roce 1982:

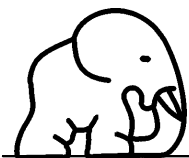
```
(Odberatele select: [:x | x rok_narozeni < 1982]).
```

- » Pravidlo - fyzická osoba nesmí mít vyplněné IČ:

```
(Odberatele select: [:x | x ic isEmpty]).
```

- » Dodavatel nesmí být fyzická osoba:

```
(Dodavatele select: [:x | x fyzicka_osoba=false]).
```

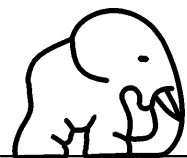


## 5. Závěr

Hlavním cílem práce bylo navrhnout model fiktivního internetového obchodu a znázornit ho relačně a objektově.

Věrné zobrazení reality by bylo velice obtížné, a proto nebyl systém uvažován v úplnosti. V praxi by bylo nutné model rozšířit. Při návrhu takového modelu by však bylo nezbytné blíže se seznámit s mechanismy internetového obchodu a nevycházet pouze z vlastních poznatků získaných z pozice zákazníka.

Pokud by byl model rozšířen a uveden v realitu, bylo by také nutné ho transformovat, neboť jeho první návrh by zřejmě nevyhovoval požadavkům kladeným na systém.



## 6. Seznam obrázků a tabulek

Obr. 1 – Entity Relationship Diagram.

Obr. 2 – Objektový model.

Tab. 1 – Atributy entit.

Tab. 2 – Atributy tříd.