Ukol 6 - trida Faktura

## Verze 1

# 2. Implementace třídy Faktura

**2.1 Atributy třídy**

Třída Faktura obsahuje následující **neměnné (immutable) atributy**:

* **Identifikace faktury**
  + cisloDokladu – číslo dokladu (String)
  + datumVystaveni – datum vystavení (LocalDate)
  + datumSplatnosti – datum splatnosti (LocalDate)
* **Příjemce faktury**
  + jmeno, prijmeni – jméno a příjmení příjemce
  + ulice, mesto, psc – adresa příjemce
  + telefon, email – kontaktní údaje
* **Finanční údaje**
  + cenaBezDPH – částka bez DPH
  + cenaSDPH – částka s DPH
  + sazbaNazev – název sazby DPH (např. "standardní")
  + sazbaHodnota – hodnota DPH v % (např. 21.0)

**2.2 Konstruktor**

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

**Jednoznačná inicializace** – Všechny údaje musí být zadány při vytvoření faktury.  
**Immutabilita** – Po vytvoření nelze údaje měnit (zajišťuje konzistenci).

# 3 Metoda toString()

Metoda vrací **formátovaný řetězec** s údaji o faktuře.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

**Příklad výstupu**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**Ladění** – Umožňuje snadné logování a debugování.  
**Čitelnost** – Formátovaný výstup pro uživatele.

# 4. Testování třídy Faktura

Byly implementovány **2 testovací případy** pomocí JUnit:

**Test 1: Ověření vytvoření faktury**

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

**Test 2: Kontrola výstupu toString()**

Une image contenant texte, capture d’écran, affichage, logiciel

Le contenu généré par l’IA peut être incorrect.

**Validace správné funkčnosti** – Ověříme, že třída funguje podle specifikace.  
**Detekce chyb** – Pokud někdo změní toString(), test selže.

# 5. Proč byla zvolena tato implementace?

1. **Immutabilita**
   * Atributy jsou final → po vytvoření nelze měnit.
   * Zabraňuje neočekávaným změnám (např. při sdílení objektu mezi vlákny).
2. **Jednoduchost a čitelnost**
   * Konstruktor přijímá všechny potřebné parametry.
   * toString() poskytuje přehledný výstup.
3. **Testovatelnost**
   * Třída je snadno testovatelná (JUnit testy).
4. **Rozšiřitelnost**
   * V budoucnu lze přidat další metody (např. getCenaSDPH()).

## **6. Ukázka použití v** Main **třídě**

Une image contenant texte, capture d’écran, logiciel, affichage

Le contenu généré par l’IA peut être incorrect.

## Verze 2

## 2. Hlavní třídy a jejich účel

**2.1 Třída CastkyDokladu**

**Účel**: Reprezentuje finanční část faktury (ceny, DPH).

**Klíčové vlastnosti:**

* Uchovává:
  + cenaBezDPH – základní částka
  + cenaSDPH – částka včetně DPH
  + sazbaNazev – název sazby ("standardní"/"snížená")
  + sazbaHodnota – procentuální hodnota DPH

**Důležité metody:**

**Testování**:

* Ověřuje se vytvoření objektu a správnost getterů
* Testuje formát výstupu toString()

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

**2.2 Třída Prijemce**

**Účel**: Reprezentuje příjemce faktury.

**Klíčové vlastnosti:**

* Uchovává kontaktní údaje:
  + Jméno, příjmení
  + Adresa (ulice, město, PSČ)
  + Kontakt (telefon, email)

**Důležitá metoda:**

**Une image contenant texte, capture d’écran, affichage, logiciel

Le contenu généré par l’IA peut être incorrect.**

**Testování**:

* Kontroluje vytvoření příjemce
* Validuje výstupní format

**2.3 Třída Faktura (s Builder patternem)**

**Účel**: Hlavní třída reprezentující fakturu.

**Klíčové změny oproti verzi 1.0:**

* Používá **kompozici** (místo primitivních typů obsahuje objekty Prijemce a CastkyDokladu)
* Implementuje **Builder pattern** pro flexibilní vytváření

Une image contenant texte, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Použiti v main

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

**Výhody Builderu**:  
 Lepší čitelnost kódu  
Možnost nastavit pouze potřebné parametry  
Neměnnost (immutability) výsledného objektu

**2.4 Utility třída ZauctujFakturu**

**Účel**: Obsahuje business logiku pro výpočet DPH.

**Klíčové metody:**

**Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.**

**Testování**:

* Ověřuje správnost výpočtů pro různé sazby DPH

## **3. Ukázka použití v praxi**

Une image contenant texte, capture d’écran, logiciel, affichage

Le contenu généré par l’IA peut être incorrect.

Výstup

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

## 4. Testování

Pro každou třídu jsou připraveny unit testy:

* CastkyDokladuTest – testuje výpočty a gettery
* PrijemceTest – ověřuje vytváření příjemce
* FakturaTest – testuje Builder a výstup
* ZauctujFakturuTest – validuje business logiku

CastkyDokladuTest

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

PrijemceTest

Une image contenant texte, capture d’écran, affichage, logiciel

Le contenu généré par l’IA peut être incorrect.

FakturaTest

Une image contenant texte, Appareils électroniques, capture d’écran, affichage

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

ZauctujFakturuTest

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Le contenu généré par l’IA peut être incorrect.

**Hlavní výhody verze 2.0**:  
Lepší organizace kódu  
Větší flexibilita  
 Snadnější údržba  
 Lepší testovatelnost