STŘEDNÍ PRŮMYSLOVÁ ŠKOLA ELEKTROTECHNICKÁ V PRAZE, JEČNÁ 30

Obor: Informační technologie - Programování a digitální technologie

---

Bankovní simulátor

Autor: *Tomáš Voborný*
Škola: *SPŠE Ječná*
Předmět: IT
Rok vytvoření: 2025

Obsah

---

1. Cíl práce

Cílem projektu je vytvořit **bankovní simulátor v jazyce Java**, který umožní uživateli spravovat svůj bankovní účet prostřednictvím konzole.

Program je určen pro studijní účely a měl by být srozumitelný, rozšiřitelný a stabilní.

Program umožňuje:

- registraci a přihlášení uživatelů,
- provádění vkladů, výběrů a převodů,
- správu účtů a zobrazování přehledů,
- ukládání a načítání dat ze souboru,
- přidělování úroků administrátorem.

---

2. Popis programu

2.1 Algoritmus

Program simuluje běžný bankovní systém v konzoli.
Po spuštění se uživatel rozhodne, zda se chce zaregistrovat nebo přihlásit.

Po úspěšném přihlášení může zadávat příkazy jako například `deposit`, `withdraw`, `transfer`, nebo `logout`.

Administrátor má navíc možnost přidat úroky všem účtům pomocí příkazu `interest`.

Data jsou uchovávána v textových souborech a načítají se při startu aplikace.
Systém zajišťuje, aby transakce byly přiřazeny správným účtům.

2.2 Klíčové objekty systému

- **Uživatel (Account)** – běžný uživatel, který může spravovat svůj účet.
- **Administrátor (AdministrationAccount)** – má přístup k pokročilým funkcím jako je přidělování úroků.
- **Systém (Console, LoginManager)** – zajišťuje komunikaci a správu příkazů.

2.3 Mechaniky

- **Registrace a přihlášení** pomocí `LoginManager`.
- **Bankovní příkazy** jako `deposit`, `withdraw`, `transfer`, `logout`, `delete`.
- **Přidělování úroku** dostupné pouze administrátorům.
- **Zobrazování informací o účtu** (`info`) a databázi (`database`).
- **Ukládání a načítání dat** ze souborů (`accounts.txt`, `transactions.txt`).

- **Textové rozhraní** s jednoduchými příkazy.

---

3. Systémové požadavky

- **Programovací jazyk**: Java 22
- **Vývojové prostředí**: IntelliJ IDEA 2024.1.1
- **Systém**: libovolný OS s JDK 22
- **Externí knihovny**: žádné – využívá pouze standardní knihovny Javy
- **Vstup/Výstup**: textová konzole (příkazový řádek)

---

4. Základní struktura

Program je objektově navržen a rozdělen do těchto hlavních částí:

- `Account` – základní třída pro uživatelské účty
- `AdministrationAccount`, `SavingsAccount`, `CheckingAccount` – speciální typy účtů
- `Transaction` – záznam transakce mezi účty
- `Database` – stará se o správu, ukládání a načítání účtů a transakcí
- `LoginManager` – zajišťuje přihlašování, registraci a přepínání účtů
- `Console` – hlavní řídící logika programu
- `Command` – jednotlivé příkazy, které uživatel může volat (`Deposit`, `Withdraw`, `Transfer`, atd.)

---

5. Testovací data

Program obsahuje jednotkové testy napsané v **JUnit 5**, které testují například:

- **TransferTest** – ověřuje správnost převodu mezi účty
- **DepositTest** – testuje úspěšné a neplatné vklady
- **WithdrawTest** – kontroluje výběry včetně neplatných částek
- **ApplyInterestTest** – ověřuje, zda se úroky správně připočítají
- **DeleteTest** – kontroluje odstranění účtu
- **LoginManagerTest** – test přihlášení, registrace a správy účtů

Tyto testy ověřují základní logiku a správné chování v běžných i hraničních situacích.

---

6. Uživatelská příručka

Program běží v **textové konzoli**.

Po spuštění se zobrazí výzva k registraci nebo přihlášení.
Poté může uživatel zadávat příkazy podle níže uvedeného seznamu:

- `deposit` – vloží částku na účet
- `withdraw` – vybere částku z účtu
- `transfer` – pošle peníze jinému uživateli
- `info` – zobrazí informace o účtu
- `interest` – administrátor přidá úroky
- `logout` – odhlásí uživatele
- `delete` – smaže účet
- `help` – zobrazí dostupné příkazy
- `exit` – ukončí program

Vstupy jsou zadávány pomocí textu.
Uživatel je vždy upozorněn, pokud zadá neplatný vstup.

---

7. Závěr

Práce na projektu byla velmi přínosná.

Během vývoje jsem si procvičil práci s objekty, kolekcemi, soubory a výjimkami.
Implementace ukládání transakcí z více účtů byla výzva, ale nakonec se ji podařilo vyřešit pomocí kontroly duplikátů.

Jednotkové testy pomohly ověřit správnost jednotlivých tříd a metod.
S výsledkem jsem spokojený a projekt je připraven k použití i dalšímu rozšiřování.
