



# BANKOVNÍ SYSTÉM

Realizace v jazyce Java

## Abstrakt

Tento dokument popisuje postup realizace mé semestrální práce na téma Bankovní systém – konzolová aplikace v jazyce Java.

Petr Boháč  
petr.bohac@tul.cz

## Obsah

Popis úlohy.....	2
Ukládání a čtení dat .....	2
Systém tříd .....	4
Bank .....	4
TerminalSession .....	4
Configuration .....	4
Komunikace s uživatelem.....	4

## Popis úlohy

Zadáním této semestrální práce bylo vytvořit plně funkční bankovní systém, který bude komunikovat s uživatelem prostřednictvím konzole. Projekt jsem trochu více rozvedl, obsahuje proto více features, než obyčejná konzolová bank, “audit log” kupříkladu.

## Ukládání a čtení dat

Data jsou na disku uložena ve formátu **JSON** (JavaScript Object Notation). Složka *data* obsahuje veškeré soubory, ze kterých se čte, nebo ukládá. Složka je dále rozdělena na tři podsložky, *accounts*, *users* a *lang*. Složky *accounts* a *users* obsahují JSON soubory korespondující k daným objektům. Soubory jsou pojmenovány ve stylu **object\_id.json** (např. *92b897bc-27d6-4c0c-b129-34f85c79578d.json*).

```
{
  "password": "$2a$10$Iky080GRXcwJDF2.cChzePM2wzqhD.rsstBub0XJQnC0pW0Xs69q",
  "preferences": {},
  "name": "The",
  "last_name": "Administrator",
  "admin": true,
  "created_at": 1653826173,
  "id": "92b897bc-27d6-4c0c-b129-34f85c79578d",
  "email": "admin@bank.com",
  "username": "admin"
}
```

Obrázek 1 - Příklad serializovaného User objektu

```
{
  "audit_log": [
    {
      "date_time": 1653827975,
      "type": "CREATION",
      "user": "92b897bc-27d6-4c0c-b129-34f85c79578d"
    }
  ],
  "balance": 0,
  "transaction_history": [],
  "name": "The Administrator's Checking account",
  "currency": "USD",
  "owners": [
    "92b897bc-27d6-4c0c-b129-34f85c79578d"
  ],
  "id": "51def32b-9f53-4437-b08f-a1d2064ab4b7",
  "type": "CHECKING_ACCOUNT"
}
```

Obrázek 2 - Příklad serializovaného Account objektu

Adresář *lang* obsahuje soubory ve formátu **YAML** (Yet Another Markup Language), obsahující data ve formátu *key, value* pair na každém řádku.

```
# Account types
savings_account: "Savings account"
checking_account: "Checking account"
retirement_account: "Retirement account"

# Messages regarding user authentication
login_prompt_username: "Username"
login_prompt_password: "Password"
login_user_not_found: "User {user} doesn't exist"
login_wrong_password: "Incorrect password! {tries} tries left."
login_wrong_password_timeout: "Incorrect password! You have been put on timeout for this account."
login_still_timeout: "You can't log into this account for another {duration}"

user_welcome: "Welcome back, {user}"
user_logout: "You have been logged out."
user_logout_exit: "You have been logged out, exiting the application."
```

Obrázek 3 - Úryvek souboru definujícího anglické rozhraní

## System tříd

Přiložený soubor *uml.png* obsahuje vizualizaci projektu ve formě UML digramu. Na diagramu jsou vidět veškeré třídy, včetně vnitřních tříd a enums, a veškeré dependency connections. Veškeré informace ke třídám, atributům tříd, metodám apod. je zakomponováno v **javadoc**.

### Bank

Hlavní třída, obsahující hlavní metodu *main(String[] args)*. Tato třída obsahuje reference ke všem načteným datum v paměti – uživatelé, bankovní účty, apod.

### TerminalSession

Tato třída implementuje *Session* čímž reprezentuje jeden typ média mezi uživatelem a programem. Mohla by existovat podobná třída, která by také implementovala *Session*, např. *GUISession* nebo *FXSession*. *TerminalSession* tedy reprezentuje komunikační médium v konzoli. Vytváří také instanci třídy *LanguageManager*, která se stará, o načítání language related dat a jejich následné zužitkování.

### Configuration

Třída *Configuration* má privátní konstruktor, tudíž není instancovatelná. Slouží pouze jako úložiště program-wide konfigurace ve veřejných statických finálních atributech.

## Komunikace s uživatelem

Poté, co se uživatel přihlásí, pokud má preferovaný jazyk, jazyk celého rozhraní je změněn na daný jazyk, a první menu je zobrazeno. V nabídce top-level menu je odhlášení, nastavení a podmenu pro account management. Adminské top-level menu je poté o něco rozšířeno (Obrázek 4).

```
Choose an option:
[1] - My accounts
[2] - Settings
[3] - Show all users
[4] - Show all accounts
[5] - Add a new user
[6] - Log out
[7] - Log out and exit
```

Obrázek 4 - Adminské top-level menu

O krok níže je account management podmenu (Obrázek 5), ze kterého je uživatel schopný otevřít si nový účet, nebo se podívat do již existujícího účtu.

```
>>> MY ACCOUNTS <<<
=====
You own/co-own 1 account(s) (4 slots left)

Adminuv account - 0.00 CZK

Choose an option:
[1] - Select account
[2] - Open a new account
[3] - Go back
```

Obrázek 5 - Account management menu

Zda-li uživatel v předchozím menu zvolí položku číslo 1, je vyzván k vybrání jednoho z jeho účtů, a poté je zobrazeno detailní menu a informace o daném účtu, jednotlivé položky jsou poměrně intuitivně popsány, tudíž nemá smysl vysvětlovat, co každá dělá.

```
>>> Adminuv account <<<
=====
Savings | Balance: 0.00 CZK | Owners: 1
Last access: admin at Jun 1, 2022, 4:57:38 PM

Choose an option:
[1] - Make a payment
[2] - View transaction history
[3] - View transaction history (ordered by amount)
[4] - Open audit log (ordered chronologically)
[5] - Settings
[6] - Close this account
[7] - Go back
```

Obrázek 6 - Detail účtu