

# Simulace portfolia – programátorský manuál

---

## 1) Programátorský manuál

### 1.1 Datový model a průtok dat

- **Aktivum** je slovník se základní strukturou:

```
{  
    'nazev': str,  
    'cena': float,           # počáteční cena  
    'vaha': float,          # cílová váha v portfoliu  
    'typ': str,              # např. 'akcie', 'dluhopis', 'zlato',  
    'hotovost'  
    'korelace': float,       # jen pro korelační model  
    'mnozstvi': float,       # doplní se po alokaci kapitálu  
    'ceny': list[float]      # doplní se během simulace (časová řada)  
}
```

- **Portfolio** je `list[aktivum]`.
- **Historie rebalancování** je `list[záznam]`, kde záznam:

```
{  
    'den': int,  
    'transakce': [  
        {'aktivum': str, 'puvodni': float, 'nove': float, 'rozdil': float,  
        'poplatek': float}  
    ],  
    'poplatky_celkem': float,  
    'hodnota_portfolia': float  
}
```

### 1.2 main.py

**Účel:** Orchestrace běhu – načtení vstupů, simulace, exporty, grafy.

**Tok:**

1. Načte konfiguraci (`nacti_konfiguraci`).
2. Pro každý soubor portfolia:
  - o `nacti_portfolio`,
  - o `vypocitej_zakladni_mnozstvi` podle `pocatecni_hodnota`,
  - o případně vygeneruje **sdílené ceny** (`generuj_sdilene_ceny`) a použije je, pokud se aktiva shodují,
  - o spustí simulaci (`simuluj_portfolio` nebo `simuluj_portfolio_sdilene`),

- vypíše statistiky (vypis\_statistiku),
  - exportuje CSV/TXT a kreslí grafy,
  - akumuluje výsledky pro pozdější porovnání.
3. Vykreslí porovnání všech portfolií (PNG i interaktivní HTML).

#### Poznámky:

- model, sdilena\_simulace, rebalanční parametry atd. jdou z konfigurace.csv.
- Při nesouladu aktiv u sdílené simulace se loguje varování a jede se nesdíleně.

#### 1.3 portfolio.py

- nacti\_portfolio(soubor='portfolio\_input.csv')
  - CSV → list aktiv; kontrola součtu vah ( $\pm 0.001$ ); ošetření chyb (soubor nenalezen, chybějící sloupec, špatná čísla).
- vypocitej\_zakladni\_mnozstvi(portfolio, pocatecni\_hodnota=1000)
  - Rozdělí kapitál podle vah, pro poslední aktivum doplní **zbytek** kvůli numerice.
  - Nastaví aktivum['mnozstvi'] u všech aktiv.
- vypocitej\_vyvoj\_portfolia(portfolio)
  - Sumarizace hodnoty přes ceny a mnozstvi (není v main.py nutně využito, protože simulace.py vrací vyvoj již přímo).

#### 1.4 konfigurace.py

- nacti\_konfiguraci(soubor='konfigurace.csv')
  - Čte CSV parametr, hodnota; konvertuje na bool/float/str (podle obsahu hodnota).
- PARAMETRY\_TYPU\_AKTIVA
  - Mapa základních ročních parametrů pro typy aktiv:

```
{
    'akcie': {'ocekavany_vynos': 0.08, 'volatilita': 0.15},
    'dluhopis': {'ocekavany_vynos': 0.03, 'volatilita': 0.06},
    'zlato': {'ocekavany_vynos': 0.035, 'volatilita': 0.08},
    'hotovost': {'ocekavany_vynos': 0.01, 'volatilita': 0.0}
}
```
- preved\_na\_denni(rocní\_vynos, rocní\_volatilita)
  - Denní drift:  $(1 + r)^{1/252} - 1$ , denní sigma: vol /  $\sqrt{252}$ .

#### 1.5 simulace.py

##### Modely denních cen:

- simuluj\_den\_nahodny(aktualni\_cena, denni\_volatilita=0.02)
  - delta ~ U(-vol, +vol); cena \*= (1 + delta).
- simuluj\_den\_typove(aktivum)

- Parametry z PARAMTRY\_TYPU\_AKTIVA[aktivum['typ']] → denní mu, sigma.
- zmena ~ N(mu, sigma); cena \*= (1 + zmena).
- simuluj\_den\_korelacni(aktivum, index\_zmena, denni\_volatilita=0.02)
  - zmena = rho \* index\_zmena + (1-rho) \* U(-vol, +vol).

### **Generátor sdílených cen:**

- generuj\_sdilene\_ceny(portfolio, pocet\_dni, model, denni\_volatilita)
  - Vrací dict{nazev\_aktiva: [cena\_0, ..., cena\_T]} pro všechna aktiva.
  - U korelačního modelu generuje i vlastní trajektorii „indexu“.

### **Simulace portfolia:**

- simuluj\_portfolio(portfolio, cilove\_vahy, pocet\_dni, rebalancovaci\_perioda, zpusob\_rebalancovani, tolerance\_vahy, transakcni\_poplatek, model, denni\_volatilita, sdilena\_simulace=False)
  - Pro každý den nasimuluje ceny dle modelu.
  - Podmínky rebalancování: perioda / odchylka / kombinace (viz níže).
  - Vrací (vyvoj, historie).
- simuluj\_portfolio\_sdilene(portfolio, cilove\_vahy, ceny\_aktiv, rebalancovaci\_perioda, zpusob\_rebalancovani, tolerance\_vahy, transakcni\_poplatek)
  - Totéž, ale bere ceny z ceny\_aktiv (již vygenerované sdílené trajektorie).

### **Rebalanční spouštěč:**

- Používá rebalancovani.je\_odchylka\_prilis\_velka – porovnává aktuální váhy (počítané z **posledních cen** v ceny[-1]) s cílovými; pokud je rozdíl > tolerance, vrací True.

### **1.6 rebalancovani.py**

- rebalancuj\_portfolio(portfolio, den, cilove\_vahy, historie, poplatek\_sazba=0.005)
  - Vypočte aktuální hodnotu portfolia v **daný den** z ceny[den].
  - Pro každé aktivum spočte **cílovou částku** = celkova\_hodnota \* cilova\_vaha.
  - **Poplatek** = |rozdíl\_castky| \* poplatek\_sazba.
  - **Skutečná částka** pro nové množství:
    - nákup: cilova\_castka - poplatek,
    - prodej: cilova\_castka + poplatek.
  - Přepočte množstvi = skutecna\_castka / aktualni\_cena.
  - Zapíše transakci do historie (včetně součtu poplatků).
- je\_odchylka\_prilis\_velka(portfolio, cilove\_vahy, tolerance=0.05)

- Porovnává váhy z **aktuálních posledních cen** vs. cíle.
- `uloz_rebalancovani_do_txt(historie, prefix="")`
  - Uloží přehled všech rebalancí do `vystupy/<prefix>/statistiky/<prefix>_rebalancovani.txt`.

## 1.7 soubory.py

- Vytváří adresářovou strukturu `vystupy/<prefix>/statistiky, grafy`.
- `uloz_transakce_do_csv(historie, prefix) → .../<prefix>_transakce.csv`
- `uloz_vyvoj_portfolia_do_csv(vyvoj, prefix) → .../<prefix>_vyvoj_portfolia.csv`
- `uloz_ceny_aktiv_do_csv(portfolio, prefix) → .../<prefix>_ceny_aktiv.csv`
- `uloz_statistiky_do_csv(vyvoj, prefix, bezrizikova_sazba=0.01) →`
  - `.../<prefix>_statistiky.csv` a `.../<prefix>_statistiky.txt`

### Poznámky k formátu:

- Číselné hodnoty jsou formátovány na rozumný počet desetinných míst.
- Ošetřeny hraniční stavy (málo dat apod.).

## 1.8 statistiky.py

- `vypis_statistiku(vyvoj, bezrizikova_sazba=0.01)` (pokračování)
  - Počítá: celkové zhodnocení, **CAGR**, denní/roční volatilitu (stdev denních výnosů), **Sharpe ratio** ( $\text{CAGR} - \text{rf}$ )/ $\text{vol\_rocni}$ , **max drawdown**, min/max hodnotu.
  - Vstupy:
    - `vyvoj: list[float]` — denní hodnoty portfolia (počáteční hodnota i hodnoty po každém dni).
    - `bezrizikova_sazba: float` — roční bezriziková sazba (default 0.01).
  - Výstup: žádny (tisk do konzole). Funkce nevrací hodnoty; metriky se zapisují do CSV/TXT v `soubory.py`.
  - Výjimky: pokud `len(vyvoj) < 2`, funkce pouze vypíše upozornění a vrátí se.

## 1.9 grafy.py

**Účel:** Generování grafů pro vizualizaci výsledků simulace. Grafy ukládají obrázky (PNG) a interaktivní HTML (Plotly).

### Hlavní funkce (signatury, vstupy, chování):

- `vykresli_ceny_aktiv(portfolio, nazev_souboru='grafy/vyvoj_ceny_aktiv.png', prefix='')`
  - `portfolio: list[dict]` — každý prvek obsahuje minimálně název a ceny (seznam cen po dnech).

- nazev\_souboru: relativní cesta v rámci podadresáře grafy/ (default grafy/vyvoj\_ceny\_aktiv.png).
  - prefix: složkový prefix pro vystupy/<prefix>/....
  - Chování: vykreslí časové řady cen všech aktiv, uloží PNG.
  - Kontroly: ověří, že portfolio není prázdné a že ceny existují.
- vykresli\_vyvoj\_portfolia(hodnoty\_portfolia, nazev\_souboru='grafy/vyvoj\_portfolia.png', prefix=' ')
  - hodnoty\_portfolia: list[float] — denní celková hodnota portfolia.
  - Vykreslí řadu a uloží PNG.
  - Kontroly: ověří, že hodnoty\_portfolia není prázdné.
- vykresli\_vyvoj\_vah(portfolio, nazev\_souboru='grafy/vyvoj\_vah.png', prefix='')
  - Vstupy jako vykresli\_ceny\_aktiv + předpoklad, že portfolio obsahuje mnozství resp. historie\_mnozství pokud chcete přesné denní váhy.
  - Doporučení: pokud chcete přesnost mezi rebalančními dny, v simulaci ukládejte snapshoty množství po každém dni (historie\_mnozství) a upravte funkci tak, aby je použila.
- vykresli\_drawdown(vyvoj, nazev\_souboru='grafy/drawdown.png', prefix=' ')
  - vyvoj: list[float].
  - Vykreslí procentuální drawdown v čase (graf v %).
- vykresli\_histogram\_vynosu(vyvoj, nazev\_souboru='grafy/histogram\_vynosu.png', prefix=' ')
  - Vypočítá denní výnosy (poměrové změny) a kreslí histogram, uloží PNG.
- vykresli\_heatmapu\_korelace(portfolio, nazev\_souboru='grafy/heatmapa\_korelace.png', prefix=' ')
  - Vytvoří DataFrame denních výnosů pro každé aktivum (pouze pokud existuje alespoň 2 ceny).
  - Vypočítá korelační matici a vykreslí heatmapu pomocí seaborn.heatmap.
- vykresli\_reálnou\_hodnotu\_portfolia(vyvoj, inflacni\_sazba=0.02, nazev\_souboru='grafy/reálna\_hodnota.png', prefix=' ')
  - Diskontuje nominální hodnoty portfolia o roční inflaci (předpokládá 252 obchodních dní).
  - Vykreslí a uloží PNG.
- vykresli\_rolling\_volatilitu(vyvoj, okno=63, nazev\_souboru='grafy/rolling\_volatilita.png', prefix=' ')
  - Používá pandas: pct\_change → rolling(window=okno).std() \* sqrt(252).
  - Uloží PNG. Okno 63 odpovídá přibližně tříměsíčnímu oknu (63 obchodních dní).
- vykresli\_vyvoj\_vice\_portfolii(vysledky, prefix=' ')
  - vysledky: dict[str, list[float]] — klíč je název portfolia, hodnota je časová řada.

- Vykreslí několik časových řad do jednoho PNG  
(vystupy/porovnani/vyvoj\_vice\_portfolii.png).
- vykresli\_vyvoj\_vice\_portfolii\_interaktivne(vysledky,  
vystup='vystupy/porovnani/vyvoj\_portfolii\_interaktivne.html')
  - Používá plotly pro vytvoření interaktivního grafu (možnost zoomu/tooltipy) a uloží HTML.

**Závislosti a další poznámky:** - Modul používá matplotlib, seaborn, pandas a plotly.

Ujistěte se, že jsou nainstalované. - Všude se vytváří adresář vystupy/<prefix>/... pokud ještě neexistuje. - Barvy a styly grafů jsou jednoduché.