

# XX.XX Závěrečný projekt - CSV adresář

---

## Závěrečný projekt - zadání

---

### Adresář CSV

1. Máme data pro databanku kontaktů ve formátu:

- `jmeno`: Křestní jméno kontaktu
- `prijmeni`: Příjmení kontaktu
- `rozliseni`: Rozlišovací znak/y pro stejná jména (např. ml., st., 1, 2, )
- `email`: Email adresa kontaktu
- `telefon`: Telefon ve tvaru +420123456789

2. Data jsou uložena v rámci CSV souboru

- sloupce mají názvy odpovídající datové struktuře kontaktu viz. bod 1.
- při spuštění programu se data načtou do "databáze"
- databáze bude `list`, nebo `dictionary`

3. Funkce dostupné v programu

- `nacist_kontakty_csv`
  - načte do zvolené "databáze" kontakty z CSV souboru
  - případně provede konverzi dat do správného formátu (nechceme existující mezery na začátku na konci, telefon nemá požadovaný formát +420123456, platný email apod.)
  - pokud v "databázi" nějaké kontakty jsou přepíší se všechny ze souboru
- `ulozit_kontakty_csv`
  - vezme aktuální "databázi" a zapíše ji do CSV souboru a přepíše ho novými aktuálními daty
- `setridit_kontakty`
  - vezme kontakty v "databázi" a seřídí je minimálně podle `prijmeni`, ideálně následně i podle `jmeno`
- `vytvorit_kontakt`
  - předejte data ve vhodném formátu (ideálně `**kwargs`, nebo i `list`, `tuple`, `dictionary`)
  - pro jednoduchost všechny parametry kontaktu jsou povinné pro zadání nového kontaktu kromě pole `rozliseni`
  - hlídejte duplicitu kontaktu podle jedinečné kombinace polí `jmeno`, `prijmeni` a `rozliseni`

- duplicitní kontakt je takový který má stejné všechny tři tyto parametry
- duplicita není povolena, takže se duplicitní kontakt nezapíše do "databáze"
- kontakt se nemusí hned zapsat do CSV souboru
- `smazat_kontakt`
  - kontakt zadaný např. indexem, nebo kombinací `jmeno`, `prijmeni` a `rozliseni`
- `najit_kontakt`
  - vyhledat libovolnou kombinací `jmeno`, `prijmeni`, `rozliseni`, `email`, `telefon`
  - mohu hledat podle jednoho nebo i více zadaných parametrů do funkce
- `detail_kontaktu`
  - funkce přijme jako parametr data kontaktu
  - zobrazí je na obrazovku jako tabulku dat

```

-----
| Jméno:      | Franta                               |
| Příjmení:   | Vopršálek                           |
| Rozlišení:  | ml.                                 | # nic pokud nezadáno
| Telefon:    | +420123456789                       |
| Email:      | franta.voprsalek@seznam.cz          |
| Index:      | 25                                  | # index listu, jiný klíč?
-----

```

- `vypsat_kontakty`
  - vezme jeden kontakt z "databáze" jeden po druhém
  - a vypíše je např. pomocí `detail_kontaktu` nebo v jiné vhodné tabulce
  - třeba i jako "excell" výstup tedy sloupce Jméno, Příjmení, Rozlišení, Telefon, Email
- `upravit_kontakt`
  - volitelná tedy nepovinná funkce
  - zadané parametry index/klíč zvoleného kontaktu a `**kwargs` nebo `dictionary` parametry které chcete změnit

## Uživatelske rozhraní

### Varianta - kontinuálně běžící program

- spustíte program
- ten vypíše na obrazovku seznam příkazů jako nápovědu uživateli odpovídající funkčnosti bodu 3. (nemusí to být příkazy ale stačí třeba stisky číslíc odpovídající příkazům)
- pak očekává zadání příkazu z klávesnice (např. stisk klávesy) a pokud příkaz vyžaduje parametry předáte mu je např. na vyžádání od programu

- plus implementujte prikaz např. "H" který znovu vypíše nápovědu v podobě seznamu příkazů

## **Varianta - class a řádková manipulace**

- existuje jeden OOP class který inicializujete v Python konzoli
- tento class má inicializační parametr cestu k souboru
- po inicializaci má interně načtenou "databázi" kontaktů ze souboru a umožní s nimi pracovat
- class má metody viz. bod 3. pro operace nad kontakty