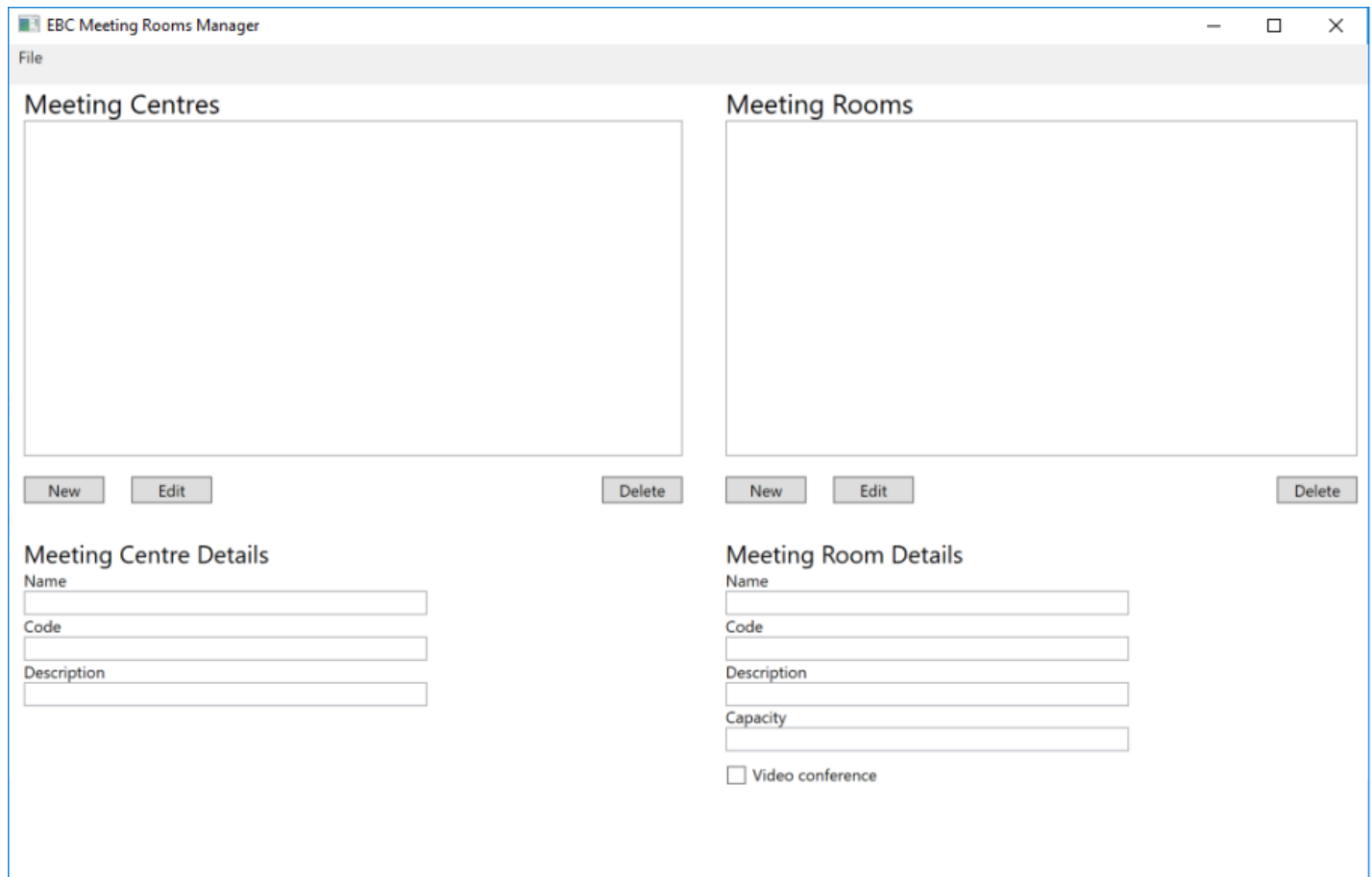


Zadat dů 2 skupině T01

## Kontext

Cílem tohoto úkolu je procvičit si programování v Javě a zejména základy práce se soubory.

Společnost vlastní několik budov, které jsou označovány jako **EBC Meeting Centres** a ty obsahují zasedací místnosti, které dále v textu budeme nazývat Meeting Rooms. Implementujte aplikaci, která umožní zpracovávat nastavení Meeting Centres a v nich Meeting Rooms. Protože jsme ještě neprobírali vytváření grafického uživatelského rozhraní, budeme improvizovat s uživatelským rozhraním pomocí zobrazeného menu možností v konzoli. Pro názornost zde máme obrázek, který by představoval grafické rozhraní naší aplikace.



Obrázek 1 - Ilustrace rozložení prvků uživatelského rozhraní aplikace pro správu místností

## Zadání

Vytvořte aplikaci podle následujícího zadání.

V aplikaci se bude pracovat s entitami Meeting Centre a Meeting Room, které mají následující vlastnosti (v závorkách je definován datový typ a popsána případná omezení):

### Meeting Centre

- Name – název budovy (řetězec o délce 2..100 znaků).
- Code – unikátní identifikátor budovy (řetězec o délce 5..50 znaků, který může obsahovat velká a malá písmena anglické abecedy a znaky: tečka, dvojtečka a podtržítko).
- Description – slovní popis budovy (řetězec o délce 10..300 znaků).
- MeetingRooms – Meeting Centre může obsahovat 0..n MeetingRooms.

### Meeting Room

- Name – název místnosti (řetězec o délce 2..100 znaků).
- Code – unikátní identifikátor místnosti (řetězec o délce 5..50 znaků, který může obsahovat velká a malá písmena anglické abecedy a znaky: tečka, dvojtečka a podtržítko).
- Description – slovní popis budovy (řetězec o délce 10..300 znaků).
- Capacity – maximální počet osob, který se pohodlně vejde do místnosti (celé číslo v rozsahu 1..100).
- VideoConference – informace zdali místnost obsahuje zařízení pro videokonferenci (ano/ne).
- MeetingCentre – Meeting Room patří právě do jednoho Meeting Centre.

## Správa Meeting Centres a Meeting Rooms

---

Aplikace umožní spravovat Meeting Centres a Meeting Rooms. Aplikace zobrazí seznam možností, které lze aktuálně provádět. Zde bude možnost zobrazení všech Meeting Centres, přičemž v seznamu bude pro každé Meeting Centre zobrazen jeho název a kód. Další možností hlavního menu bude přidání nového Meeting Centra a následně import dat ze souboru, který bude popsán níže. Budou zde ještě dvě možnosti, kterými se aplikace ukončí. Jedna z nich bude navíc obsahovat uložení dat do souboru tak, aby si aplikace provedené změny při příštím otevření pamatovala.

Ve chvíli, kdy uživatel vybere možnost zobrazení všech Meeting Center, aplikace vypíše na obrazovku seznam a pod tento seznam vypíše opět menu možností, kam se uživatel může ve svém počínání ubírat. Toto bude obecné pravidlo pro jakýkoliv krok v aplikaci, že pod příslušným zobrazením bude opět možnost volby, i kdyby tam měla být třeba jen jedna volba a to zpět na předchozí krok.

Po vylistování všech center by tu měla být možnost zobrazení detailu konkrétního meeting centra, kde budou k němu zobrazeny všechny informace (v ukázce jde o sekci Meeting Centre Details + počet Meeting Rooms) a zároveň možnosti pro editaci, zobrazení seznamu Meeting Rooms nebo přidání nové místnosti.

Zároveň má uživatel možnost založit nové Meeting Centre, kde máte možnost zobrazit všechny potřebné atributy a následně se ptát na každý z nich, samozřejmě zde bude i možnost přidat nové místnosti.

Další možností bude smazat Meeting Centre. Ve chvíli, kdy uživatel tuto funkčnost spustí, tak bude zobrazen dotaz, ve kterém uživatel potvrdí nebo stornuje, že chce opravdu Meeting Centre smazat. Pokud bude Meeting Centre smazáno, pak budou smazány i do něj patřící Meeting Rooms.

S Meeting Rooms lze provádět stejné operace jako s Meeting Centres, tedy je vybrat, čímž se zobrazí detailní informace, upravit a smazat stejným způsobem jako Meeting Centres.


## Import dat

---

Aplikace umožní importovat data. V hlavním menu po zvolení Importu dat nás aplikace vyzve k zadání cesty k souboru, který chceme importovat. Soubor bude uložen na našem lokálním úložišti. Pokud jsou v aplikaci nějaká data, pak budou smazána a následně budou do aplikace nahrána data nová z importovaného souboru. Soubor s daty je ve formátu CSV, přičemž platí následující:

- První řádek obsahuje řetězec MEETING\_CENTRES.
- Následuje 1..n řádků s informacemi o jednotlivých budovách. Tyto informace jsou na řádku v následujícím pořadí: Name, Code, Description.

- Následuje řádek, který obsahuje řetězec `MEETING_ROOMS`.
- Následuje 1..m řádků s informacemi o jednotlivých zasedacích místnostech. Tyto informace jsou na řádku v následujícím pořadí: `Name,Code,Description,Capacity,VideoConference,MeetingCentreCode`(kód Meeting Centre, do kterého Meeting Room patří).
- Data na řádku jsou oddělena čárkou, přičemž samotná data nebudou obsahovat čárku.
- Předpokládejte, že data jsou v korektním formátu, tzn., že např. nechybí na některém řádku hodnota a zároveň data neporušují omezení popsána u Meeting Centre a Meeting Room výše. Zároveň předpokládejte, že v souboru jsou kódy budov a místností unikátní a že místnosti nejsou přiřazeny neexistujícím budovám. Nemusíte se tedy zabývat validacemi tohoto souboru.
- Používá kódování UTF-8.

Ukázkový soubor s daty pro import naleznete v příloze tohoto artefaktu:  [ImportData.csv](#)

## Ukládání dat aplikace

---

V kapitole Import dat jsme se zabývali funkcí, která umožní do aplikace vložit data. Import typicky uživatelé využijí při prvním spuštění aplikace. Nicméně aplikace si musí sama uchovávat svá data. Tzn., že úpravy, které uživatel provede, ať už u Meeting Centres nebo u Meeting Rooms, musejí být po ukončení aplikace příslušným krokem z hlavního menu (Exit and Save) uloženy a při opětovném spuštění načteny.

Uložení dat opět provedte do souboru formátu csv. Ideálně byste měli dodržet stejný formát, jaký má soubor, ze kterého se data importují. Pokud ho dodržíte, ušetříte si práci s načítáním dat při inicializaci aplikace. Soubor se bude nacházet ve stejném adresáři jako spuštěná aplikace, název tohoto souboru můžete mít pevně dán ve zdrojovém kódu.

Při spuštění aplikace si rovnou načtete data z tohoto souboru, pokud již tento soubor existuje a obsahuje potřebná data. Načtení při startu vám pak usnadní vypisování dat uživateli a nutnosti při každém zobrazení nějakých informací znovu parsovat soubor. Budeme předpokládat, že do něj přistupujete pouze vy z vaší aplikace. Pokud by to byl soubor dat sdílený, museli bychom řešit více komplikací.

## Obecné požadavky

---

Všechny uživatelské vstupy musí být ošetřeny tak, že uživatelem zadané hodnoty nezpůsobí neočekávanou chybu (tj. např. ukončení programu). Kromě toho by zadávání údajů mělo být co nejkomfortnější (např. pokud uživatel zadá jeden údaj špatně, program by ho o tom měl informovat co nejdříve a umožnit mu změnu jen tohoto jediného údaje a nenutit ho znovu zadávat celý formulář).

V přiloženém souboru naleznete předpřipravený projekt, kde jsou již vytvořeny objekty a hlavní menu aplikace. Na vás bude toto řešení rozšířit na kompletní rozsah zadání. Primárně o práci s načítáním a ukládáním souboru.

- Třída `MainController` je hlavní třídou, která spouští aplikaci a zobrazí hlavní menu.
- `MeetingController` je třída obsahující základní metody, které jsou volané z hlavního menu týkající se zobrazení a správu dat.
- Třída `FileParser` bude obsahovat práci se soubory.
- Třída `Choices` obsahuje obsluhu menu.
- V balíčku `model` jsou obsaženy třídy jednotlivých objektů.

## Hodnocení

---

Body budou udělovány za:

- Funkčnost kódu - 70%
- Styl psaní kódu (např. dodržování objektivně-orientovaného přístupu), jeho srozumitelnost a dodržování Java konvencí - 30%

Zcela nefunkční, přesto srozumitelný kód neznamená automaticky udělení 30% bodů (výše uvedené hodnoty v procentech proto považujte za spíše orientační).

## Řešení

---

Odeslat řešení domácího úkolu