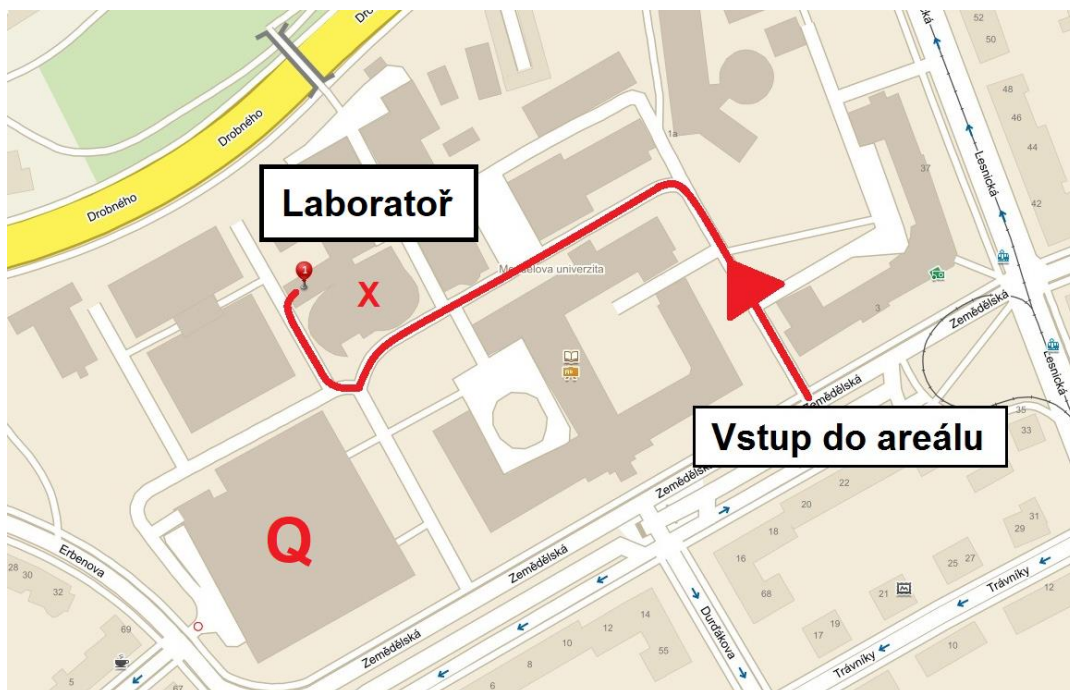


Praktická cvičení z Algoritmizace v laboratoři řízení kolejových vozidel

Během posledního výukového týdne semestru proběhnou v rámci předmětu Algoritmizace praktická cvičení, která nahradí cvičení „standardní“. Praktická cvičení budou probíhat v [laboratoři řízení kolejových vozidel](#).



Příprava na praktické cvičení

V rámci přípravy na praktické cvičení vypracujete níže zadané úkoly. Cílem těchto úkolů bude připravit funkční programy, se kterými budete schopni ovládat jednotlivé prvky modulového kolejiště.

Výčet úloh, které lze na kolejišti řešit:

1. práce s kolejištěm

- a. přehazování výhybek
- b. ovládání rozpojovačů
- c. zjišťování stavu obsazenosti traťových úseků
- d. stavění jízdních cest

2. práce s vozidly

- a. rozjetí/zastavení lokomotivy
- b. změna směru jízdy lokomotivy
- c. nastavení funkcí lokomotivy (zapnutí světel, zapnutí zvuků, houkání...)

Základní princip řešení výše zmíněných úloh:

1. Ze serveru kolejiště získáme údaje o všech prvcích daného typu.
2. Získané údaje uložíme do souboru.
3. Vyhledáme v souboru *id* prvku, který odpovídá námi zadanému *názvu*.
4. Ze serveru kolejiště získáme detailní informace o prvku daného *id*.
5. Získané detailní informace opět uložíme do souboru.
6. V souboru vyhledáme *stav/hodnotu* požadovaného atributu.
7. Na základě tohoto zjištění posléze rozhodneme o *hodnotě*, kterou budeme chtít nově k atributu přiřadit.
8. Novou *hodnotu* atributu následně odešleme na server kolejiště.
9. Změna hodnoty se projeví na kolejišti (např. dojde k přehození výhybky).

Řídící programy:

- **vyhybky.cpp** – přehodí výhybku, jejíž *název* je zadán jako 1. parametr z příkazového řádku, do opačného směru
- **vozidla_jizda.cpp** – nastaví lokomotivě, jejíž *název* je zadán jako 1. parametr z příkazového řádku, *rychlostní stupeň*, který je zadán jako 2. parametr z příkazového řádku (nejnižší *stupeň* – 0, nejvyšší *stupeň* – 28)
- **vozidla_smer.cpp** – nastaví lokomotivě, jejíž *název* je zadán jako 1. parametr z příkazového řádku, opačný směr jízdy
- **jizdni_cesty.cpp** – postaví jízdní cestu, jejíž *název* je zadán jako 1. parametr z příkazového řádku

Podrobné popisy jednotlivých programů naleznete přímo v příslušných souborech. Všechny tyto programy můžete plně převzít a dále upravovat.

Úkoly na praktické cvičení

Úkol 1:

Všechny řídicí programy využívají již připravené moduly. Jediný modul, který budete muset doimplementovat, je *mzjisti.cpp*. Podstatou 1. úkolu je implementace funkce, která na základě zadaného *názvu* prvku vyhledá odpovídající hodnotu *id*.

Při implementaci funkce můžete použít libovolný postup (možnou variantou řešení je např. i čtení souboru a následná úprava řetězce znaků). Podrobné zadání naleznete přímo v souboru *mzjisti.cpp* v adresáři *ukol_1*.

Úkol 1 – náhradní varianta řešení:

Tento úkol je určen pouze studentům, kterým se nepodařilo splnit „základní“ zadání úkolu 1. Pokud se Vám nepodařilo dokončit modul *mzjisti.cpp*, můžete řešit jeho náhradu – potřebná data (*id* prvku) zjistíte „ručním“ dohledáním v datech získaných přímo ze serveru kolejiště. Vaším úkolem tak bude příprava upravených řídicích programů.

Každý program bude rozdělen na 2 části. První část získá ze serveru kolejiště přehled prvků daného druhu (výhybky, lokomotivy, jízdní cesty). Druhou část bude představovat upravená verze řídicího programu, kdy místo *názvu* prvku budeme rovnou zadávat „ručně“ dohledané *id* prvku. Výsledná funkce upravených řídicích programů zůstane nezměněna.

Podrobné zadání naleznete v souboru *nahradni_zadani.txt* v adresáři *nahradni_ukol*.

Úkol 2:

V souboru *vozidla_jizda.cpp* naleznete program, který dokáže rozjet lokomotivu na zadaný rychlostní stupeň. Existuje však také možnost zapínat a vypínat různé funkce lokomotivy – zvuk, světla... Podstatou 2. úkolu bude příprava programu, který dokáže ovládat funkce lokomotivy. Podrobnější zadání úkolu naleznete v souboru *zadani.txt* v adresáři *ukol_2*.

Závěr

Do svého adresáře na akele si před zahájením cvičení přichystejte již zkompileované řídicí programy, jejichž funkčnost bude právě v průběhu cvičení ověřována. Pro kompilaci programů budete mimo jiné potřebovat již připravené součásti, které naleznete v adresáři *zakladni_soucasti*.