

Práce se sítí

Motivace

Pro účely modelování a simulace provozu elektrizační rozvodné soustavy potřebujeme znát její topologii. Důležité je umět identifikovat izolované ostrovy a to jak fyzicky izolované, tj. dva uzly sítě nejsou fyzicky propojeny, tak i galvanické, tj. dva uzly sítě mohou být fyzicky propojeny, ale nejsou propojeny vodičem (rozepnutý spínač). V základní variantě budeme uvažovat pouze fyzické propojení. Zadání lze rozšířit o galvanické propojení, kdy by ve vstupním souboru byla navíc ještě informace o galvanickém propojení (u definice hran grafu).

Režimy programu

Program funguje ve dvou režimech: S grafickým uživatelským rozhraním (GUI) nebo na příkazové řádce. Tyto dva režimy jsou odlišeny přepínačem *--gui*. Pokud je tento přepínač při spuštění přítomen, pak bude program pracovat v režimu s GUI a všechny další vstupy budou zadávány v tomto GUI. V režimu příkazové řádky má program na vstupu jeden nepovinný parametr, který reprezentuje cestu k souboru s daty. Pokud není tento vstup uveden, program načítá data ze standardního vstupu.

Vstupy

Příkazová řádka

Vstupní data obsahují dvě sekce oddělené jedním prázdným řádkem. První sekce je výčet názvů uzlů sítě (název nesmí obsahovat čárku), kdy každý uzel je uveden na samostatném řádku. Druhá sekce obsahuje hrany grafu, které reprezentují propojení mezi jednotlivými uzly ve formátu **uzelA,uzelB**. V případě čtení hodnot ze standardního vstupu je načítání ukončeno koncem souboru - řeší operační systém, program nemusí speciálně řešit rozdíly mezi Linux a Windows. Klávesové zkratky pro konec souboru jsou na příkazové řádce následující:

- Linux *Ctrl + d*
- Windows *Ctrl + z*

Příklad vstupního souboru:

A
B
C
D
E

A,B
B,C
E,D

GUI

Pro vstupní data budou v GUI připraveny dva grafické prvky pro vložení seznamu uzlů a seznamu hran grafu (viz. předchozí kapitola) a tlačítko pro spuštění vyhledávání.

Výstupy

Příkazová řádka

Program vypíše na standardní výstup následující informace:

- Celkový počet ostrovů.
- Prázdný řádek.
- Výpis jednotlivých ostrovů, kdy jména uzlů pro daný ostrov budou na jednom řádku oddělená čárkou a každý ostrov bude na vlastním řádku. Ostrovy budou seřazeny podle jejich velikosti vzestupně.

Příklad výstupu (jména uzlů nemusí být ve stejném pořadí jako na vstupu):

2

E,D
C,A,B

GUI

Pro výstup bude sloužit samostatný grafický prvek zobrazující nalezené ostrovy.

Další vlastnosti programu

Program bude validovat vstupní parametr. Je-li parametr přítomen, musí reprezentovat buďto přepínač `--gui` nebo existující soubor s právy ke čtení. V případě porušení výše uvedených pravidel, program vypíše na standardní chybový výstup hlášku a návod na použití programu.

Dále bude probíhat validace jména uzlu - nesmí obsahovat čárku a definice hrany - vždy obsahuje dvě jména existujících uzlů oddělených čárkou. Pokud program zjistí neplatné vstupy, oznámí to uživateli tak, aby bylo jasné, co konkrétně porušilo validační pravidla. V režimu s GUI tak učiní modálním oknem s chybovou hláškou. V režimu příkazové řádky program vypíše chybovou hlášku na **standardní chybový výstup** (stderr), nikoliv na standardní výstup (stdout).

Tipy pro implementaci

Programujte proti rozhraním. Oddělte logiku práce se vstupy a výstupy od samotného algoritmu pro nalezení ostrovů. Pište unit testy. Pamatujte, že každá třída by měla mít pouze jeden úkol. Používejte kontejnerové typy (kolekce).

Začněte návrhem rozhraní pro algoritmus. Poté jej implementujte. Implementaci si otestujte pomocí unit testů (přes rozhraní). Až poté, co budete mít hotovu tuto část, řešte zpracování vstupů, výstupů, napojení na GUI a ostatní věci.

Tam kde to dává smysl (např. spojení vstupu, výstupu a algoritmu), zkuste použít návrhový

vzor Dependency Injection (DI) - pokud moje třída závisí na jiné, pak se na ni odkazují pouze přes její rozhraní a nevytvářím přímo její instanci, ale tuto instanci si nechám předat přes konstruktor nebo přes setter. Tento styl programování významně pomáhá udržet čistý objektový návrh a zlepšuje testovatelnost kódu.

Když něco nevíte, ptejte se!