

Prezentace projektu II.

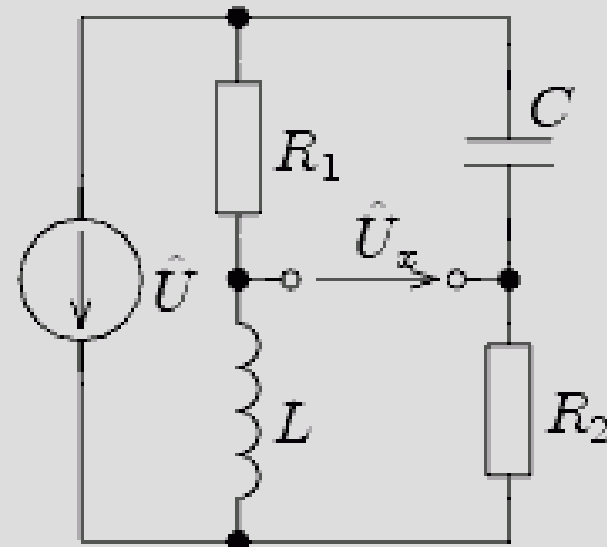
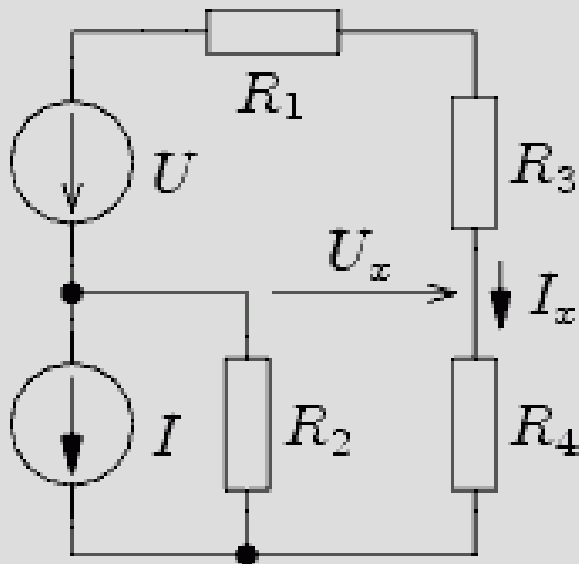
X36ASS

Řešení lineárních elektrických sítí

Jan Skalický

Motivace

- Elektrická síť s lineárními aktivními a pasivními dipóly, pracujících v ustáleném stacionárním nebo harmonickém stavu



Hotovo

- Jádro analytického řešícího softwaru
- Vstupem uživatele je popis obvodu ve tvaru soustavy součástek a jejich propojení
- Výstupem programu je nalezení neznámých obvodových veličin (napětí, proud, výkon)

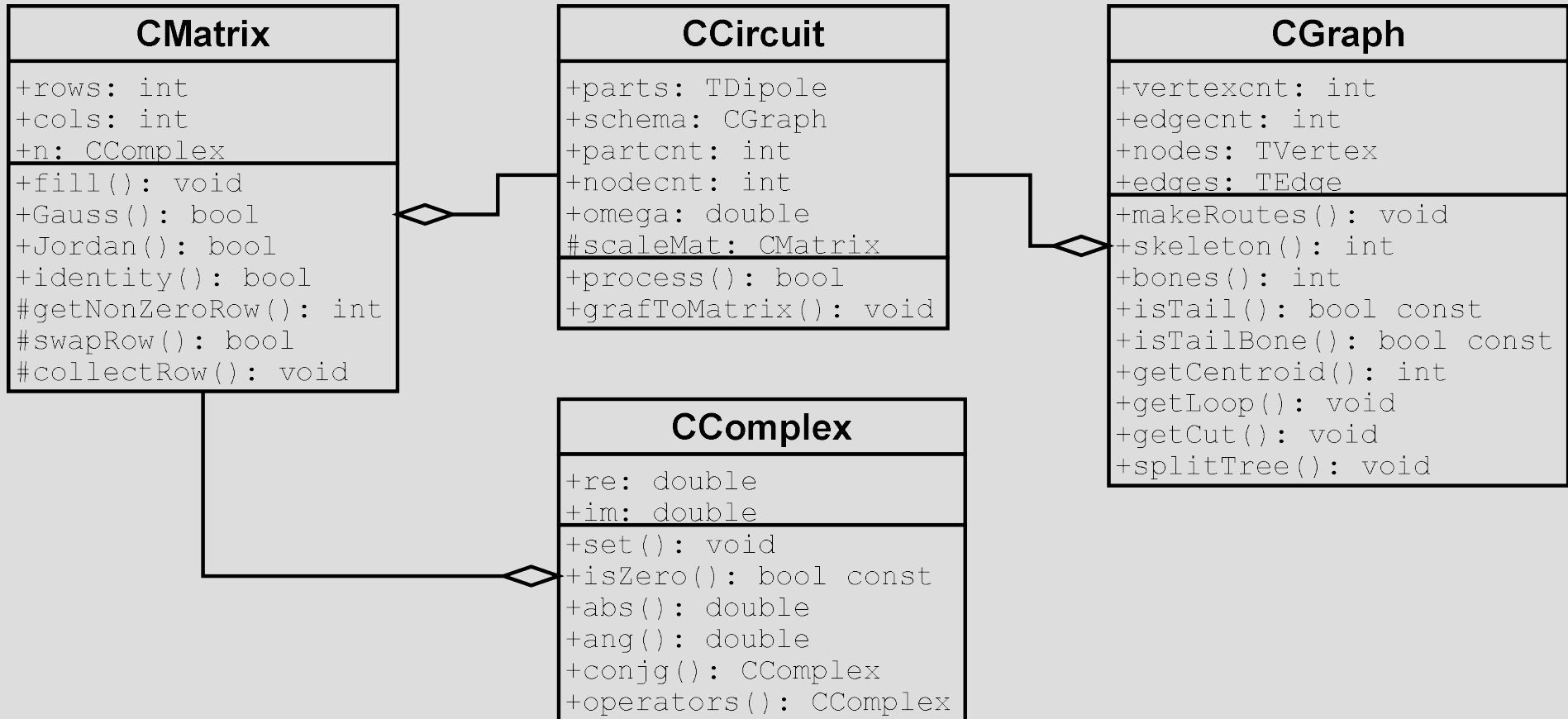
Implementace v C++

- Dekompozice do 4 tříd a 4 struktur
- Data grafu schématu obvodu:
 - pole vrcholů (TVertex *nodes) zastupující uzly obvodu
 - pole směrů (TRoute *routes) k sousedním vrcholům přes incidentní hrany
 - počet směrů (int routecnt)
 - počet obsažených hran kostry (int bones)
 - značka pro trasovací účely (int tag)
 - pole hran (TEdge *edges) zastupující větve obvodu
 - indexy vrcholů (int vex1, vex2), které spojuje
 - je součástí kostry? (boolean bone)
 - značka pro trasovací účely (int tag)

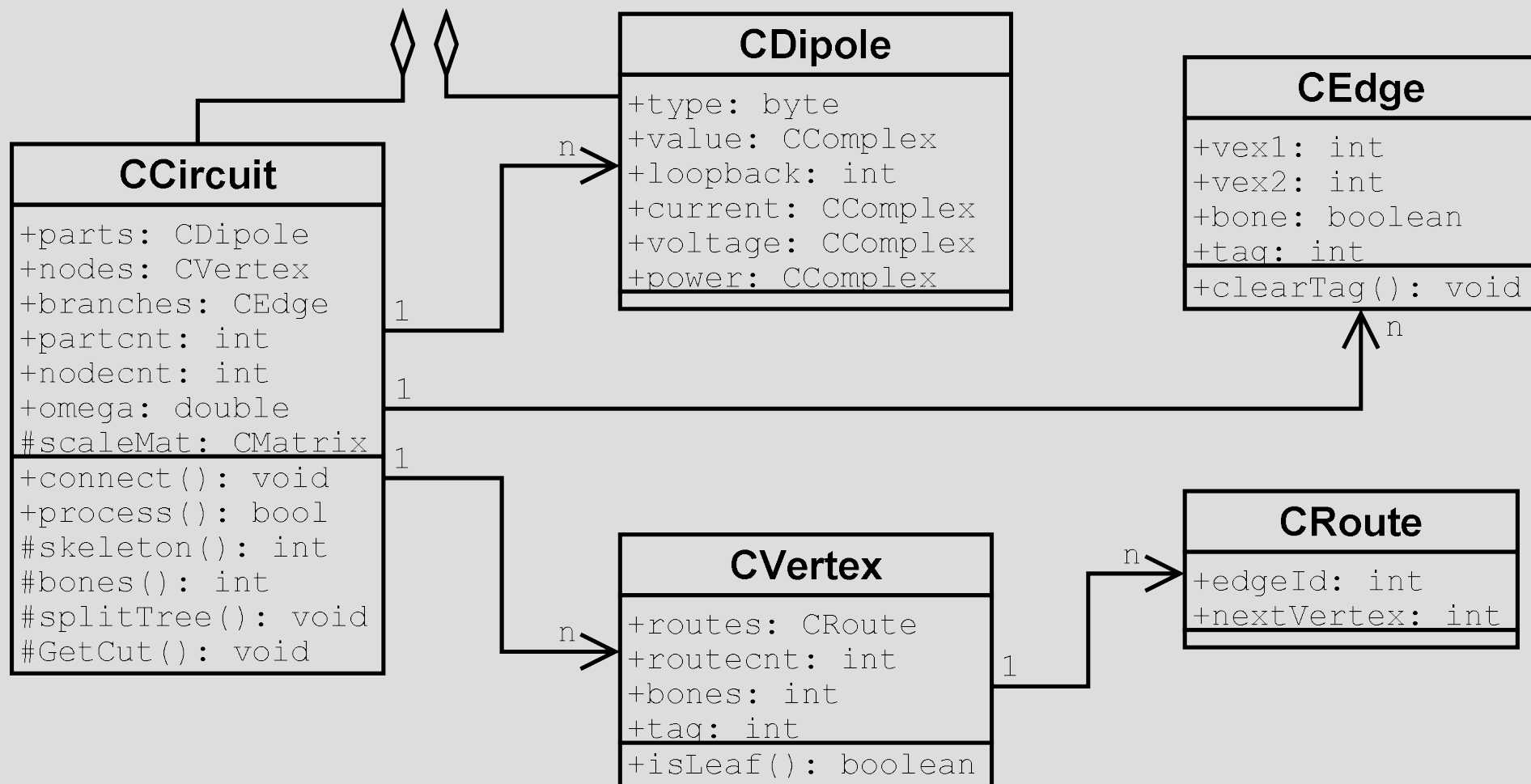
Implementace v Javě – rozdíly

- Rozdílná dekompozice vzhledem k striktně objektovému přístupu – 7 tříd:
 - vrcholy, hrany a směry jsou jednoduché třídy
- Odlišná alokace polí objektů (alokace pointerů místo volání konstruktorů v C++)
- Omezení v některých konstruktech jazyka (for, goto) – programátorský aspekt

Class diagram



Class diagram – CGraph v Javě



Algoritmy pro analýzu grafu

- Jsou instančními metodami třídy CCircuit (pracují s objekty nodes, branches)
 - nalezení a označení kostry grafu s co nejvíce listy; metoda vrací počet segmentů
 - spočtení obsažených hran kostry pro každý vrchol grafu; metoda vrací počet nalezených listů
 - nalezení nezávislé smyčky a naplnění adekvátního řádku matice její rovnicí
 - nalezení nezávislého řezu naplnění adekvátního řádku matice jeho rovnicí

Řešení komplexní matice

- heterogenní soustava s komplexní maticí
 - 1 rovnice napětí ve smyčce pro nezávislou hranu
 - 1 rovnice proudu v uzlu pro kostrovou hranu
 - 1 rovnice charakteristiky součástky pro každou hranu
- => jednoznačné řešení (nebo žádné)
 - Gaussova eliminace
 - Jordanova eliminace
 - převod na jednotkovou matici
- vyjmutí výsledků, spočtení příkonů, výstup

Porovnání architektur

C++

- Použití struktur (lépe optimalizovatelné)
- Rychlejší běh, ale je nutno rekompilovat pro platformu
- Vlastní alokace paměti (předvídatelné chování)

Java

- Všechno je třída (více systémové, režie)
- Není nutno rekompilovat, ale vyžaduje JRE
- Spotřeba paměti ovlivněna implementací GC

Zhodnocení

- Program pracuje spolehlivě
 - byl několikrát testován pro různé typy obvodů a všechny výsledky vždy souhlasily se správným řešením
- Důraz při výběru zpracovávaného tématu byl kladen i na jeho využitelnost v praxi
- Nejvíce se povedlo vymyslet princip činnosti metody pro hledání nezávislých řezů

Uživatelské rozhraní

- Ukázková konzolová aplikace (testovací main):

```
pocet vetvi: 2
pocet uzlu : 2
vetev 0 od uzlu: 1
vetev 0 do uzlu: 0
typ soucastky (0=J, 1=U, 2=I, 3=R): 1
hodnota soucastky: 10
vetev 1 od uzlu: 1
vetev 1 do uzlu: 0
typ soucastky (0=J, 1=U, 2=I, 3=R): 3
hodnota soucastky: 2

vetev 0: v kostre=1, napeti=10.000000, proud=-5.000000, vykon=-50.000000
vetev 1: v kostre=0, napeti=10.000000, proud=5.000000, vykon=50.000000
```

Prezentace projektu II.

Děkuji za pozornost