

Evidence výpočetní techniky II.

Termín odevzdání:	28.04.2013 23:59:59
Pozdní odevzdání s penalizací:	12.05.2013 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)
Hodnocení:	4.4000
Max. hodnocení:	4.0000 (bez bonusů)
Odevzdaná řešení:	2 / 20 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání)
Nápovědy:	0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úloha je rozšířením jednodušší úlohy Evidence výpočetní techniky o možnosti virtualizace a cloudových řešení. Úkolem je navrhnout a implementovat sadu tříd, které budou simulovat evidenci počítačového vybavení firmy. Konkrétně budeme ukládat informace o sítích (CNetwork), počítačích (CComputer), jejich procesorech (CCPU), pamětech (CMemory) a discích (CDisk). Oproti jednodušší úloze půjde modelovat virtualizované sítě a počítače, tedy počítače a sítě, které jsou obsažené v jiném počítači. Vytvořené počítače a sítě navíc půjde kopírovat.

Úkol je zaměřen na návrh tříd, kde bude využito dědičnosti, polymorfismu a abstraktních metod. Pokud jsou tyto OOP prostředky použité racionálně, není implementace příliš dlouhá. Naopak, pokud provedete návrh špatně, bude se Vám kód opakovat a implementační soubor bude velký. Zkuste identifikovat základní třídu a vhodně z ní děděním odvodte podtřídy.

Třídy a jejich rozhraní:

CNetwork

reprezentuje síť. Její rozhraní musí obsahovat:

- konstruktor se jménem sítě,
- destruktorka, kopírující konstruktor a operátor = (pokud je potřeba vlastní implementace),
- metodu AddComputer, kterou lze přidávat další počítač do sítě,
- metodu FindComputer, která vrátí odkaz na nalezený počítač zadaného jména, NULL, pokud nenalezne. Pozor, hledá se i ve virtuálních počítačích, které např. běží uvnitř jiného počítače, který je připojen k této síti,
- metodu FindNetwork, která vrátí odkaz na nalezenou síť zadaného jména, NULL, pokud nenalezne. Pozor, hledá se i ve virtuálních sítích, které např. běží uvnitř jiného počítače, který je připojen k této síti,
- výstupní operátor, který zobrazí strom počítačů a komponent, jako v ukázce. Počítače jsou vypsané v pořadí přidávání.

CComputer

reprezentuje počítač. Její rozhraní musí obsahovat:

- konstruktor s parametrem jména počítače
- destruktorka, kopírující konstruktor a operátor = (pokud je potřeba vlastní implementace),
- metoda AddComponent, která přidá další komponentu počítače,
- metoda AddAddress, která přidá další adresu počítače (řetězec),
- metodu FindComputer, která vrátí odkaz na nalezený počítač zadaného jména, NULL, pokud nenalezne. Výsledkem může být tento počítač sám nebo i virtuální počítač, který běží na tomto počítači,
- metodu FindNetwork, která vrátí odkaz na nalezenou síť zadaného jména, NULL, pokud nenalezne. Pozor, hledá se i ve virtuálních sítích, které např. běží uvnitř tohoto počítače,
- metoda Duplicate(remap), která vytvoří kopii tohoto počítače. Vytváří se identická kopie (až na případné změny jména), včetně případných virtualizovaných počítačů/sítí. Při vytváření kopie metoda navíc změní jména všech kopírovaných počítačů a sítí podle parametru (remap). Tento parametr je tvořen dvojicí řetězců (původní jméno, nové jméno). Pokud se jméno kopírovaného počítače/sítě objeví v parametru remap jako nějaké původní jméno, bude nahrazeno. Pokud původní jméno v parametru obsaženo není, bude ponecháno beze změny.
- operátor pro výstup, který zobrazí přidělené adresy a komponenty počítače, jako v ukázce. Ve výpisu jsou nejprve uvedené adresy (v pořadí zadání) a za nimi komponenty (v pořadí přidávání).

CCPU

reprezentuje CPU. Její rozhraní musí obsahovat:

- konstruktor s parametrem počtu jader (celé číslo) a frekvencí (celé číslo v MHz),
- destruktorka, kopírující konstruktor a operátor = (pokud je potřeba vlastní implementace).

CMemory

reprezentuje RAM. Její rozhraní musí obsahovat:

- konstruktor s parametrem velikosti paměti (celé číslo v MiB),
- destruktork, kopírující konstruktor a operátor = (pokud je potřeba vlastní implementace).

CDisk

reprezentuje úložiště. Její rozhraní musí obsahovat:

- konstruktor s parametry typu disku (symbolická konstanta SSD nebo MAGNETIC deklarovaná ve třídě) a velikosti disku (celé číslo v GiB),
- destruktork, kopírující konstruktor a operátor = (pokud je potřeba vlastní implementace),
- metodu AddPartition, která přidá informaci o rozdělení disku. Metoda bude mít dva parametry - velikost parcely v GiB a její identifikaci (řetězec). Výpis parcel je v pořadí zadávání.

Odevzdávejte zdrojový kód se implementací tříd CNetwork, CComputer, CCPU, CMemory a CDisk. Do odevzdávaného souboru zahrňte všechny potřebné podpůrné deklarace. Části vkládání hlaviček a Vaše testy ponechte v bloku podmíněného překladu, jak je ukázáno níže.

Rozhraní implementovaných tříd:

```
#ifndef __PROGTEST__
#include <iostream>
#include <iomanip>
#include <string>
#include <vector>
#include <map>
#include <set>
#include <list>
#include <algorithm>
using namespace std;
#endif /* __PROGTEST__ */
```

```
class CComputer ...
class CNetwork ...
class CCPU ...
class CMemory ...
class CDisk ...
```

Ukázka použití tříd:

```
template <int _CNT>
map<string,string> makeMap ( const pair<string,string> (&data)[_CNT] )
{
    return map<string,string> ( data, data + _CNT );
}

...
```

```
CNetwork n ( "FIT network" );
n . AddComputer (
    CComputer ( "progtest.fit.cvut.cz" ) .
        AddAddress ( "147.32.232.142" ) .
        AddComponent ( CCPU ( 8, 2400 ) ) .
        AddComponent ( CCPU ( 8, 1200 ) ) .
        AddComponent ( CDisk ( CDisk::MAGNETIC, 1500 ) .
            AddPartition ( 50, "/" ) .
            AddPartition ( 5, "/boot" ) .
            AddPartition ( 1000, "/var" ) ) .
        AddComponent ( CDisk ( CDisk::SSD, 60 ) .
            AddPartition ( 60, "/data" ) ) .
        AddComponent ( CMemory ( 2000 ) ) .
        AddComponent ( CMemory ( 2000 ) ) ) .
    AddComputer (
        CComputer ( "edux.fit.cvut.cz" ) .
            AddAddress ( "147.32.232.158" ) .
            AddComponent ( CCPU ( 4, 1600 ) ) .
            AddComponent ( CMemory ( 4000 ) ) .
            AddComponent ( CDisk ( CDisk::MAGNETIC, 2000 ) .
                AddPartition ( 100, "/" ) .
                AddPartition ( 1900, "/data" ) ) ) .
```

```

    AddComputer (
        CComputer ( "imap.fit.cvut.cz" ) .
            AddAddress ( "147.32.232.238" ) .
            AddComponent ( CCPU ( 4, 2500 ) ) .
            AddAddress ( "2001:718:2:2901::238" ) .
            AddComponent ( CMemory ( 8000 ) ) );
cout << n;
/*
--8<-----8<-----8<-----8<-----8<-----8<--
Network: FIT network
+-Host: progtest.fit.cvut.cz
| +-147.32.232.142
| +-CPU, 8 cores @ 2400MHz
| +-CPU, 8 cores @ 1200MHz
| +-HDD, 1500 GiB
| | +-[0]: 50 GiB, /
| | +-[1]: 5 GiB, /boot
| | \-[2]: 1000 GiB, /var
| +-SSD, 60 GiB
| | \-[0]: 60 GiB, /data
| +-Memory, 2000 MiB
| \-Memory, 2000 MiB
+-Host: edux.fit.cvut.cz
| +-147.32.232.158
| +-CPU, 4 cores @ 1600MHz
| +-Memory, 4000 MiB
| \-HDD, 2000 GiB
| | +-[0]: 100 GiB, /
| | \-[1]: 1900 GiB, /data
\ -Host: imap.fit.cvut.cz
| +-147.32.232.238
| +-2001:718:2:2901::238
| +-CPU, 4 cores @ 2500MHz
| \-Memory, 8000 MiB
--8<-----8<-----8<-----8<-----8<-----8<--
*/
CNetwork x = n;
CComputer * c = x . FindComputer ( "imap.fit.cvut.cz" );
cout << *c;
/*
--8<-----8<-----8<-----8<-----8<-----8<--
Host: imap.fit.cvut.cz
+-147.32.232.238
+-2001:718:2:2901::238
+-CPU, 4 cores @ 2500MHz
\ -Memory, 8000 MiB
--8<-----8<-----8<-----8<-----8<-----8<--
*/
c -> AddComponent ( CDisk ( CDisk::MAGNETIC, 1000 ) .
    AddPartition ( 100, "system" ) .
    AddPartition ( 200, "WWW" ) .
    AddPartition ( 700, "mail" ) );
cout << x;
/*
--8<-----8<-----8<-----8<-----8<-----8<--
Network: FIT network
+-Host: progtest.fit.cvut.cz
| +-147.32.232.142
| +-CPU, 8 cores @ 2400MHz
| +-CPU, 8 cores @ 1200MHz
| +-HDD, 1500 GiB
| | +-[0]: 50 GiB, /
| | +-[1]: 5 GiB, /boot
| | \-[2]: 1000 GiB, /var
| +-SSD, 60 GiB
| | \-[0]: 60 GiB, /data
| +-Memory, 2000 MiB
| \-Memory, 2000 MiB
+-Host: edux.fit.cvut.cz

```

```

| +-147.32.232.158
| +-CPU, 4 cores @ 1600MHz
| +-Memory, 4000 MiB
| \-HDD, 2000 GiB
|   +-[0]: 100 GiB, /
|   \-[1]: 1900 GiB, /data
\-Host: imap.fit.cvut.cz
  +-147.32.232.238
  +-2001:718:2:2901::238
  +-CPU, 4 cores @ 2500MHz
  +-Memory, 8000 MiB
  \-HDD, 1000 GiB
    +-[0]: 100 GiB, system
    +-[1]: 200 GiB, WWW
    \-[2]: 700 GiB, mail
--8<-----8<-----8<-----8<-----8<---
*/
cout << n;
/*
--8<-----8<-----8<-----8<-----8<---
Network: FIT network
+-Host: progtest.fit.cvut.cz
| +-147.32.232.142
| +-CPU, 8 cores @ 2400MHz
| +-CPU, 8 cores @ 1200MHz
| +-HDD, 1500 GiB
|   +-[0]: 50 GiB, /
|   +-[1]: 5 GiB, /boot
|   \-[2]: 1000 GiB, /var
| +-SSD, 60 GiB
|   \-[0]: 60 GiB, /data
| +-Memory, 2000 MiB
| \-Memory, 2000 MiB
+-Host: edux.fit.cvut.cz
| +-147.32.232.158
| +-CPU, 4 cores @ 1600MHz
| +-Memory, 4000 MiB
| \-HDD, 2000 GiB
|   +-[0]: 100 GiB, /
|   \-[1]: 1900 GiB, /data
\-Host: imap.fit.cvut.cz
  +-147.32.232.238
  +-2001:718:2:2901::238
  +-CPU, 4 cores @ 2500MHz
  \-Memory, 8000 MiB
--8<-----8<-----8<-----8<-----8<---
*/
n . AddComputer ( CComputer ( "vm01.fit.cvut.cz" ) ) .
  AddAddress ( "147.32.232.232" ) .
  AddComponent ( CCPU ( 32, 4000 ) ) .
  AddComponent ( CMemory ( 32768 ) ) .
  AddComponent ( CDisk ( CDisk::MAGNETIC, 3000 ) ) .
  AddPartition ( 500, "SYSTEM" ) .
  AddPartition ( 2000, "DATA" ) .
  AddPartition ( 500, "BACKUP" ) ) .
  AddComponent ( CNetwork ( "dummy00" ) ) .
  AddComputer ( CComputer ( "testing.fit.cvut.cz" ) ) .
    AddAddress ( "192.168.1.1" ) .
    AddComponent ( CCPU ( 1, 300 ) ) .
    AddComponent ( CMemory ( 256 ) ) ) ) .
  AddComponent ( CMemory ( 16384 ) ) .
  AddComponent ( CNetwork ( "vnet00" ) ) );
cout << n;
/*
--8<-----8<-----8<-----8<-----8<---
Network: FIT network
+-Host: progtest.fit.cvut.cz
| +-147.32.232.142
| +-CPU, 8 cores @ 2400MHz

```

```

| +-CPU, 8 cores @ 1200MHz
| +-HDD, 1500 GiB
| | +-[0]: 50 GiB, /
| | +-[1]: 5 GiB, /boot
| | \-[2]: 1000 GiB, /var
| +-SSD, 60 GiB
| | \-[0]: 60 GiB, /data
| +-Memory, 2000 MiB
| \-Memory, 2000 MiB
+-Host: edux.fit.cvut.cz
| +-147.32.232.158
| +-CPU, 4 cores @ 1600MHz
| +-Memory, 4000 MiB
| \-HDD, 2000 GiB
| | +-[0]: 100 GiB, /
| | \-[1]: 1900 GiB, /data
+-Host: imap.fit.cvut.cz
| +-147.32.232.238
| +-2001:718:2:2901::238
| +-CPU, 4 cores @ 2500MHz
| \-Memory, 8000 MiB
\ -Host: vm01.fit.cvut.cz
  +-147.32.232.232
  +-CPU, 32 cores @ 4000MHz
  +-Memory, 32768 MiB
  +-HDD, 3000 GiB
  | +-[0]: 500 GiB, SYSTEM
  | +-[1]: 2000 GiB, DATA
  | \-[2]: 500 GiB, BACKUP
  +-Network: dummy00
  | \-Host: testing.fit.cvut.cz
  | | +-192.168.1.1
  | | +-CPU, 1 cores @ 300MHz
  | | \-Memory, 256 MiB
  +-Memory, 16384 MiB
  \-Network: vnet00
--8<----8<----8<----8<----8<----8<--
*/
CNetwork * vmnet = n . FindNetwork ( "vnet00" );
vmnet -> AddComputer ( CComputer ( "monitor1.fit.cvut.cz" ) .
    AddAddress ( "147.32.232.254" ) .
    AddComponent ( CCPU ( 2, 1500 ) ) .
    AddComponent ( CMemory ( 4096 ) ) .
    AddComponent ( CDisk ( CDisk::MAGNETIC, 750 ) .
        AddPartition ( 100, "root" ) .
        AddPartition ( 600, "log" ) ) );

cout << *vmnet;
/*
--8<----8<----8<----8<----8<----8<--
Network: vnet00
\ -Host: monitor1.fit.cvut.cz
  +-147.32.232.254
  +-CPU, 2 cores @ 1500MHz
  +-Memory, 4096 MiB
  \-HDD, 750 GiB
  | +-[0]: 100 GiB, root
  | \-[1]: 600 GiB, log
--8<----8<----8<----8<----8<----8<--
*/
cout << n;
/*
--8<----8<----8<----8<----8<----8<--
Network: FIT network
+-Host: progtest.fit.cvut.cz
| +-147.32.232.142
| +-CPU, 8 cores @ 2400MHz
| +-CPU, 8 cores @ 1200MHz
| +-HDD, 1500 GiB
| | +-[0]: 50 GiB, /

```

```

| | +-[1]: 5 GiB, /boot
| | \- [2]: 1000 GiB, /var
| +-SSD, 60 GiB
| | \- [0]: 60 GiB, /data
| +-Memory, 2000 MiB
| \-Memory, 2000 MiB
+-Host: edux.fit.cvut.cz
| +-147.32.232.158
| +-CPU, 4 cores @ 1600MHz
| +-Memory, 4000 MiB
| \-HDD, 2000 GiB
|   +-[0]: 100 GiB, /
|   \- [1]: 1900 GiB, /data
+-Host: imap.fit.cvut.cz
| +-147.32.232.238
| +-2001:718:2:2901::238
| +-CPU, 4 cores @ 2500MHz
| \-Memory, 8000 MiB
\ -Host: vm01.fit.cvut.cz
  +-147.32.232.232
  +-CPU, 32 cores @ 4000MHz
  +-Memory, 32768 MiB
  +-HDD, 3000 GiB
  | +-[0]: 500 GiB, SYSTEM
  | +-[1]: 2000 GiB, DATA
  | \- [2]: 500 GiB, BACKUP
  +-Network: dummy00
  | \-Host: testing.fit.cvut.cz
  |   +-192.168.1.1
  |   +-CPU, 1 cores @ 300MHz
  |   \-Memory, 256 MiB
  +-Memory, 16384 MiB
  \-Network: vnet00
    \-Host: monitor1.fit.cvut.cz
      +-147.32.232.254
      +-CPU, 2 cores @ 1500MHz
      +-Memory, 4096 MiB
      \-HDD, 750 GiB
        +-[0]: 100 GiB, root
        \- [1]: 600 GiB, log
--8<-----8<-----8<-----8<-----8<--
*/
static const pair<string,string> remap1 [] =
{
    pair<string,string>("monitor1.fit.cvut.cz","monitor2.fit.cvut.cz")
};
vmnet -> AddComputer ( n . FindComputer ( "monitor1.fit.cvut.cz" )
                      -> Duplicate ( makeMap ( remap1 ) ) );
cout << *vmnet;
/*
--8<-----8<-----8<-----8<-----8<--
Network: vnet00
+-Host: monitor1.fit.cvut.cz
| +-147.32.232.254
| +-CPU, 2 cores @ 1500MHz
| +-Memory, 4096 MiB
| \-HDD, 750 GiB
|   +-[0]: 100 GiB, root
|   \- [1]: 600 GiB, log
\ -Host: monitor2.fit.cvut.cz
  +-147.32.232.254
  +-CPU, 2 cores @ 1500MHz
  +-Memory, 4096 MiB
  \-HDD, 750 GiB
    +-[0]: 100 GiB, root
    \- [1]: 600 GiB, log
--8<-----8<-----8<-----8<-----8<--
*/
static const pair<string,string> remap2 [] =

```

```

{
    pair<string,string>("monitor1.fit.cvut.cz","monitor3.fit.cvut.cz"),
    pair<string,string>("monitor2.fit.cvut.cz","monitor4.fit.cvut.cz"),
    pair<string,string>("vm01.fit.cvut.cz","vm02.fit.cvut.cz")
};
n . AddComputer ( n . FindComputer ( "vm01.fit.cvut.cz" )
                    -> Duplicate ( makeMap ( remap2 ) ) );
cout << n;
/*
--8<----8<----8<----8<----8<----8<--
Network: FIT network
+-Host: progtest.fit.cvut.cz
| +-147.32.232.142
| +-CPU, 8 cores @ 2400MHz
| +-CPU, 8 cores @ 1200MHz
| +-HDD, 1500 GiB
| | +-[0]: 50 GiB, /
| | +-[1]: 5 GiB, /boot
| | \-[2]: 1000 GiB, /var
| +-SSD, 60 GiB
| | \-[0]: 60 GiB, /data
| +-Memory, 2000 MiB
| \-Memory, 2000 MiB
+-Host: edux.fit.cvut.cz
| +-147.32.232.158
| +-CPU, 4 cores @ 1600MHz
| +-Memory, 4000 MiB
| \-HDD, 2000 GiB
| | +-[0]: 100 GiB, /
| | \-[1]: 1900 GiB, /data
+-Host: imap.fit.cvut.cz
| +-147.32.232.238
| +-2001:718:2:2901::238
| +-CPU, 4 cores @ 2500MHz
| \-Memory, 8000 MiB
+-Host: vm01.fit.cvut.cz
| +-147.32.232.232
| +-CPU, 32 cores @ 4000MHz
| +-Memory, 32768 MiB
| +-HDD, 3000 GiB
| | +-[0]: 500 GiB, SYSTEM
| | +-[1]: 2000 GiB, DATA
| | \-[2]: 500 GiB, BACKUP
+-Network: dummy00
| \-Host: testing.fit.cvut.cz
| | +-192.168.1.1
| | +-CPU, 1 cores @ 300MHz
| | \-Memory, 256 MiB
+-Memory, 16384 MiB
\ -Network: vnet00
    +-Host: monitor1.fit.cvut.cz
    | +-147.32.232.254
    | +-CPU, 2 cores @ 1500MHz
    | +-Memory, 4096 MiB
    | \-HDD, 750 GiB
    | | +-[0]: 100 GiB, root
    | | \-[1]: 600 GiB, log
    \-Host: monitor2.fit.cvut.cz
    | +-147.32.232.254
    | +-CPU, 2 cores @ 1500MHz
    | +-Memory, 4096 MiB
    | \-HDD, 750 GiB
    | | +-[0]: 100 GiB, root
    | | \-[1]: 600 GiB, log
\ -Host: vm02.fit.cvut.cz
    +-147.32.232.232
    +-CPU, 32 cores @ 4000MHz
    +-Memory, 32768 MiB
    +-HDD, 3000 GiB

```

```

| +-[0]: 500 GiB, SYSTEM
| +-[1]: 2000 GiB, DATA
| \-[2]: 500 GiB, BACKUP
+-Network: dummy00
| \-Host: testing.fit.cvut.cz
|   +-192.168.1.1
|   +-CPU, 1 cores @ 300MHz
|   \-Memory, 256 MiB
+-Memory, 16384 MiB
\--Network: vnet00
  +-Host: monitor3.fit.cvut.cz
  | +-147.32.232.254
  | +-CPU, 2 cores @ 1500MHz
  | +-Memory, 4096 MiB
  | \-HDD, 750 GiB
  |   +-[0]: 100 GiB, root
  |   \-[1]: 600 GiB, log
  \-Host: monitor4.fit.cvut.cz
    +-147.32.232.254
    +-CPU, 2 cores @ 1500MHz
    +-Memory, 4096 MiB
    \-HDD, 750 GiB
      +-[0]: 100 GiB, root
      \-[1]: 600 GiB, log
--8<-----8<-----8<-----8<-----8<--
*/
static const pair<string,string> remap3 [] =
{
  pair<string,string>("monitor3.fit.cvut.cz","monitor5.fit.cvut.cz"),
  pair<string,string>("monitor4.fit.cvut.cz","monitor6.fit.cvut.cz"),
  pair<string,string>("vm02.fit.cvut.cz","vm03.fit.cvut.cz"),
  pair<string,string>("vnet00","vnet01")
};
vmnet -> AddComputer ( n . FindComputer ( "vm02.fit.cvut.cz" )
                      -> Duplicate ( makeMap ( remap3 ) ) );
cout << *vmnet;
/*
--8<-----8<-----8<-----8<-----8<-----8<--
Network: vnet00
+-Host: monitor1.fit.cvut.cz
| +-147.32.232.254
| +-CPU, 2 cores @ 1500MHz
| +-Memory, 4096 MiB
| \-HDD, 750 GiB
|   +-[0]: 100 GiB, root
|   \-[1]: 600 GiB, log
+-Host: monitor2.fit.cvut.cz
| +-147.32.232.254
| +-CPU, 2 cores @ 1500MHz
| +-Memory, 4096 MiB
| \-HDD, 750 GiB
|   +-[0]: 100 GiB, root
|   \-[1]: 600 GiB, log
\--Host: vm03.fit.cvut.cz
  +-147.32.232.232
  +-CPU, 32 cores @ 4000MHz
  +-Memory, 32768 MiB
  +-HDD, 3000 GiB
  | +-[0]: 500 GiB, SYSTEM
  | +-[1]: 2000 GiB, DATA
  | \-[2]: 500 GiB, BACKUP
  +-Network: dummy00
  | \-Host: testing.fit.cvut.cz
  |   +-192.168.1.1
  |   +-CPU, 1 cores @ 300MHz
  |   \-Memory, 256 MiB
  +-Memory, 16384 MiB
  \--Network: vnet01
    +-Host: monitor5.fit.cvut.cz

```



```

| +-147.32.232.254
| +-CPU, 2 cores @ 1500MHz
| +-Memory, 4096 MiB
| \-HDD, 750 GiB
|   +-[0]: 100 GiB, root
|   \-[1]: 600 GiB, log
\--Host: monitor6.fit.cvut.cz
  +-147.32.232.254
  +-CPU, 2 cores @ 1500MHz
  +-Memory, 4096 MiB
  \-HDD, 750 GiB
    +-[0]: 100 GiB, root
    \-[1]: 600 GiB, log
--8<----8<----8<----8<----8<----8<----8<--
*/
static const pair<string,string> remap4 [] =
{
  pair<string,string>("monitor1.fit.cvut.cz","monitor11.fit.cvut.cz"),
  pair<string,string>("monitor2.fit.cvut.cz","monitor12.fit.cvut.cz"),
  pair<string,string>("vm01.fit.cvut.cz","vm11.fit.cvut.cz"),
  pair<string,string>("vm03.fit.cvut.cz","vm13.fit.cvut.cz"),
  pair<string,string>("vnet00","vnet10"),
  pair<string,string>("vnet01","vnet11")
};
n . FindComputer ( "vm02.fit.cvut.cz" )
-> FindNetwork ( "vnet00" )
-> AddComputer ( n . FindComputer ( "vm01.fit.cvut.cz" )
               -> Duplicate ( makeMap ( remap4 ) ) );
cout << n;
/*
--8<----8<----8<----8<----8<----8<----8<--
Network: FIT network
+-Host: progtest.fit.cvut.cz
| +-147.32.232.142
| +-CPU, 8 cores @ 2400MHz
| +-CPU, 8 cores @ 1200MHz
| +-HDD, 1500 GiB
| | +-[0]: 50 GiB, /
| | +-[1]: 5 GiB, /boot
| | \-[2]: 1000 GiB, /var
| +-SSD, 60 GiB
| | \-[0]: 60 GiB, /data
| +-Memory, 2000 MiB
| \-Memory, 2000 MiB
+-Host: edux.fit.cvut.cz
| +-147.32.232.158
| +-CPU, 4 cores @ 1600MHz
| +-Memory, 4000 MiB
| \-HDD, 2000 GiB
|   +-[0]: 100 GiB, /
|   \-[1]: 1900 GiB, /data
+-Host: imap.fit.cvut.cz
| +-147.32.232.238
| +-2001:718:2:2901::238
| +-CPU, 4 cores @ 2500MHz
| \-Memory, 8000 MiB
+-Host: vm01.fit.cvut.cz
| +-147.32.232.232
| +-CPU, 32 cores @ 4000MHz
| +-Memory, 32768 MiB
| +-HDD, 3000 GiB
| | +-[0]: 500 GiB, SYSTEM
| | +-[1]: 2000 GiB, DATA
| | \-[2]: 500 GiB, BACKUP
+-Network: dummy00
| \-Host: testing.fit.cvut.cz
| | +-192.168.1.1
| | +-CPU, 1 cores @ 300MHz
| | \-Memory, 256 MiB

```

```
+--Memory, 16384 MiB
\--Network: vnet00
  +-Host: monitor1.fit.cvut.cz
  | +-147.32.232.254
  | +-CPU, 2 cores @ 1500MHz
  | +-Memory, 4096 MiB
  | \--HDD, 750 GiB
  |   +-[0]: 100 GiB, root
  |   \--[1]: 600 GiB, log
+-Host: monitor2.fit.cvut.cz
| +-147.32.232.254
| +-CPU, 2 cores @ 1500MHz
| +-Memory, 4096 MiB
| \--HDD, 750 GiB
|   +-[0]: 100 GiB, root
|   \--[1]: 600 GiB, log
\--Host: vm03.fit.cvut.cz
  +-147.32.232.232
  +-CPU, 32 cores @ 4000MHz
  +-Memory, 32768 MiB
  +-HDD, 3000 GiB
  | +-[0]: 500 GiB, SYSTEM
  | +-[1]: 2000 GiB, DATA
  | \--[2]: 500 GiB, BACKUP
+-Network: dummy00
| \--Host: testing.fit.cvut.cz
|   +-192.168.1.1
|   +-CPU, 1 cores @ 300MHz
|   \--Memory, 256 MiB
+-Memory, 16384 MiB
\--Network: vnet01
  +-Host: monitor5.fit.cvut.cz
  | +-147.32.232.254
  | +-CPU, 2 cores @ 1500MHz
  | +-Memory, 4096 MiB
  | \--HDD, 750 GiB
  |   +-[0]: 100 GiB, root
  |   \--[1]: 600 GiB, log
  \--Host: monitor6.fit.cvut.cz
    +-147.32.232.254
    +-CPU, 2 cores @ 1500MHz
    +-Memory, 4096 MiB
    \--HDD, 750 GiB
      +-[0]: 100 GiB, root
      \--[1]: 600 GiB, log
\--Host: vm02.fit.cvut.cz
  +-147.32.232.232
  +-CPU, 32 cores @ 4000MHz
  +-Memory, 32768 MiB
  +-HDD, 3000 GiB
  | +-[0]: 500 GiB, SYSTEM
  | +-[1]: 2000 GiB, DATA
  | \--[2]: 500 GiB, BACKUP
+-Network: dummy00
| \--Host: testing.fit.cvut.cz
|   +-192.168.1.1
|   +-CPU, 1 cores @ 300MHz
|   \--Memory, 256 MiB
+-Memory, 16384 MiB
\--Network: vnet00
  +-Host: monitor3.fit.cvut.cz
  | +-147.32.232.254
  | +-CPU, 2 cores @ 1500MHz
  | +-Memory, 4096 MiB
  | \--HDD, 750 GiB
  |   +-[0]: 100 GiB, root
  |   \--[1]: 600 GiB, log
+-Host: monitor4.fit.cvut.cz
| +-147.32.232.254
```

```

| +-CPU, 2 cores @ 1500MHz
| +-Memory, 4096 MiB
| \-HDD, 750 GiB
|   +-[0]: 100 GiB, root
|   \-[1]: 600 GiB, log
\--Host: vm11.fit.cvut.cz
  +-147.32.232.232
  +-CPU, 32 cores @ 4000MHz
  +-Memory, 32768 MiB
  +-HDD, 3000 GiB
  | +-[0]: 500 GiB, SYSTEM
  | +-[1]: 2000 GiB, DATA
  | \-[2]: 500 GiB, BACKUP
  +-Network: dummy00
  | \-Host: testing.fit.cvut.cz
  |   +-192.168.1.1
  |   +-CPU, 1 cores @ 300MHz
  |   \-Memory, 256 MiB
  +-Memory, 16384 MiB
  \-Network: vnet10
    +-Host: monitor11.fit.cvut.cz
    | +-147.32.232.254
    | +-CPU, 2 cores @ 1500MHz
    | +-Memory, 4096 MiB
    | \-HDD, 750 GiB
    |   +-[0]: 100 GiB, root
    |   \-[1]: 600 GiB, log
    +-Host: monitor12.fit.cvut.cz
    | +-147.32.232.254
    | +-CPU, 2 cores @ 1500MHz
    | +-Memory, 4096 MiB
    | \-HDD, 750 GiB
    |   +-[0]: 100 GiB, root
    |   \-[1]: 600 GiB, log
    \-Host: vm13.fit.cvut.cz
      +-147.32.232.232
      +-CPU, 32 cores @ 4000MHz
      +-Memory, 32768 MiB
      +-HDD, 3000 GiB
      | +-[0]: 500 GiB, SYSTEM
      | +-[1]: 2000 GiB, DATA
      | \-[2]: 500 GiB, BACKUP
      +-Network: dummy00
      | \-Host: testing.fit.cvut.cz
      |   +-192.168.1.1
      |   +-CPU, 1 cores @ 300MHz
      |   \-Memory, 256 MiB
      +-Memory, 16384 MiB
      \-Network: vnet11
        +-Host: monitor5.fit.cvut.cz
        | +-147.32.232.254
        | +-CPU, 2 cores @ 1500MHz
        | +-Memory, 4096 MiB
        | \-HDD, 750 GiB
        |   +-[0]: 100 GiB, root
        |   \-[1]: 600 GiB, log
        \-Host: monitor6.fit.cvut.cz
          +-147.32.232.254
          +-CPU, 2 cores @ 1500MHz
          +-Memory, 4096 MiB
          \-HDD, 750 GiB
            +-[0]: 100 GiB, root
            \-[1]: 600 GiB, log
--8<-----8<-----8<-----8<-----8<--
*/

```

Poznámky

- Používejte operátory pro přetypování (`dynamic_cast`) s rozmyslem. Referenční implementace v sobě nemá žádné přetypování ani žádné použití RTTI. Obecně, RTTI, `dynamic_cast` a `typeid` vedou k více větvenému kódu, který je hůře čitelný a hůře rozšiřitelný. Navrhněte třídy tak, abyste si vystačili s polymorfismem.
- Všimněte si, že v ukázce chybí hlavičkový soubor `typeinfo`, tedy operátor `typeid` nelze používat.
- Vaše řešení musí používat třídy, třídy musí tvořit hierarchii, dědičnost a polymorfismus musí být použité. V této úloze je použití dědičení a polymorfismu vhodné, navíc, testovací prostředí odmítne řešení, které by dědičení, polymorfismus a dynamicky vázané metody nevyužívalo (takové řešení bude odmítnuto na chybě při kompilaci).

☐ Referenční řešení

2	18.04.2013 14:27:50	Download
Stav odevzdání:	Ohodnoceno	
Hodnocení:	4.4000	

- **Hodnotitel: automat**
 - Program zkompilován
 - Test 'Zakladni test podle ukazky': Úspěch
 - Dosáženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.001 s (limit: 5.000 s)
 - Využití paměti: 13420 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test navrhů tríd': Úspěch
 - Dosáženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.000 s (limit: 4.999 s)
 - Využití paměti: 13420 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test nahodnými hodnotami': Úspěch
 - Dosáženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 1.109 s (limit: 4.999 s)
 - Využití paměti: 14424 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test kopírujícího konstruktora': Úspěch
 - Dosáženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.042 s (limit: 3.890 s)
 - Využití paměti: 15512 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test operatoru=': Úspěch
 - Dosáženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.043 s (limit: 3.848 s)
 - Využití paměti: 15512 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test nahodnými daty + kontrola práce s pamětí': Úspěch
 - Dosáženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 7.334 s (limit: 15.000 s)
 - Využití paměti: 21832 KiB (limit: 29366 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Všechny paměťové bloky byly uvolněny - ok.
 - Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00)
- Celkové procentní hodnocení: 100.00 %
- Bonus za včasné odevzdání: 0.40
- Celkem bodů: 1.00 * (4.00 + 0.40) = 4.40

		Celkem	Průměr	Maximum	Jméno funkce
SW metrik:	Funkce:	57	--	--	--
	Řádek kódu:	394	6.91 ± 5.38	22	<code>print(ostream &,string)</code>
	Cykloomatická složitost:	74	1.30 ± 1.56	6	<code>FindNetwork(string)</code>

1	18.04.2013 14:18:38	Download
Stav odevzdání:	Ohodnoceno	
Hodnocení:	4.4000	

- **Hodnotitel: automat**

- Program zkompileován
- Test 'Základní test podle ukázky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.001 s (limit: 5.000 s)
 - Využití paměti: 13416 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test návrhu tříd': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.000 s (limit: 4.999 s)
 - Využití paměti: 13416 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnými hodnotami': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 1.239 s (limit: 4.999 s)
 - Využití paměti: 14432 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test kopírujícího konstruktora': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.029 s (limit: 3.760 s)
 - Využití paměti: 14744 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test operátoru=': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.041 s (limit: 3.731 s)
 - Využití paměti: 15220 KiB (limit: 17267 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnými daty + kontrola práce s pamětí': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 6.514 s (limit: 15.000 s)
 - Využití paměti: 22116 KiB (limit: 29366 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Všechny paměťové bloky byly uvolněné - ok.
- Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00)
- Celkové procentní hodnocení: 100.00 %
- Bonus za včasné odevzdání: 0.40
- Celkem bodů: 1.00 * (4.00 + 0.40) = 4.40

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	59	--	--	--
	Řádek kódu:	499	8.46 ± 13.38	102	main()
	Cyklomatická složitost:	75	1.27 ± 1.54	6	FindNetwork(string)