Evidence výpočetní techniky I.

Termín odevzdání:	**28.04.2019 23:59:59**
Pozdní odevzdání s penalizací:	**30.06.2019 23:59:59** (Penále za pozdní odevzdání: 100.0000 %)
Hodnocení:	**4.4000**
Max. hodnocení:	**4.0000** (bez bonusů)
Odevzdaná řešení:	1 / 20 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání)
Nápovědy:	0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

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`).

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 odvoďte podtřídy.

Třídy a jejich rozhraní:

`CNetwork`
: reprezentuje síť. Její rozhraní musí obsahovat:

- konstruktor se jménem sítě,
- destruktor, 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 nebo neplatný ukazatel, pokud jej nenalezne,
- 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
- destruktor, kopírující konstruktor a operátor = (pokud je potřeba vlastní implementace),
- metoda `AddComponent`, která přidá další komponentu počítače,
- metoda `AddAddress`, která přidá další adresu počítače (řetězec),
- operátor pro výstup, který zobrazí přidělená 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),
- destruktor, 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),
- destruktor, 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),
- destruktor, 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 v přiloženém archivu.

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 dědění a polymorfismu vhodné, navíc, testovací prostředí odmítne řešení, které by dědění, polymorfismus a dynamicky vázané metody nevyužívalo (takové řešení bude odmítnuto na chybě při kompilaci).
- Výstup je ve formě "stromu", byť v této úloze je strom nejvýše tříúrovňový. Všimněte si, že svislé čáry jsou zobrazené pouze tam, kde mají smysl. Dále si všimněte, že poslední odbočka má podobu jednoho zpětného lomítka.

Správné řešení této úlohy, které splní závazné testy na 100%, může být odevzdáno k code review.

Vzorová data:		**Download**
Odevzdat:	Choose File No file chosen	**Odevzdat**

Referenční řešení

1	**21.04.2019 20:53:45**	**Download**
Stav odevzdání:	Ohodnoceno	
Hodnocení:	4.4000	

- **Hodnotitel: automat**
  - Program zkompilován
  - Test 'Zakladni test podle ukazky': Úspěch
    - Dosaženo: 100.00 %, požadováno: 100.00 %
    - Celková doba běhu: 0.000 s (limit: 3.000 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - Test 'Test navrhu trid': Úspěch
    - Dosaženo: 100.00 %, požadováno: 100.00 %
    - Celková doba běhu: 0.000 s (limit: 3.000 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - Test 'Test nahodnymi hodnotami': Úspěch
    - Dosaženo: 100.00 %, požadováno: 50.00 %
    - Celková doba běhu: 0.022 s (limit: 3.000 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - Test 'Test kopirujiciho konstruktoru': Úspěch
    - Dosaženo: 100.00 %, požadováno: 50.00 %
    - Celková doba běhu: 0.001 s (limit: 2.978 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - Test 'Test operatoru=': Úspěch
    - Dosaženo: 100.00 %, požadováno: 50.00 %
    - Celková doba běhu: 0.001 s (limit: 2.977 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - Test 'Test nahodnymi daty + kontrola prace s pameti': Úspěch
    - Dosaženo: 100.00 %, požadováno: 50.00 %
    - Celková doba běhu: 0.063 s (limit: 4.000 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - 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

SW metriky:

	Celkem	Průměr	Maximum	Jméno funkce
Funkce:	**28**	--	--	--
Řádek kódu:	**277**	9.89 ± 27.85	153	`main`
Cyklomatická složitost:	**51**	1.82 ± 1.87	8	`CComputer::print`