

## Obecné informace

- Zadání musíte splnit v časovém limitu 3 hodiny, který začíná po vysvětlení zadání a zodpovězení dotazů k zadání. Je možné pokládat dotazy i během testu, ale tak, aby nerušily vaše kolegy při práci.
- Vaše řešení bude testováno ručně. Není nutné dbát na detaily ze zadání (např. co se týče přesného obsahu chybových/informačních hlášení). Hodnotí se implementace hlavních konceptů úlohy a jejich funkčnost.
- Je dovoleno používat tištěné i elektronické materiály, pokud to není přímo řešení dané úlohy. Není dovolena jakákoliv forma komunikace s kýmkoliv z jakéhokoli důvodu.
- Pokud budete mít hotovo dříve než uplyne časový limit, je možné se pokusit řešení odevzdat dříve. Pokud v řešení budou chyby, je možné je (v rámci časového limitu) opravit.
- Po úspěšném splnění testu zašlete všechny zdrojové soubory na email s předmětem s předmětem **Test 14.1.2020** na adresu zkoušejícího — [kliber@d3s.mff.cuni.cz](mailto:kliber@d3s.mff.cuni.cz).

## Zadání — Kontrola přiřaditelnosti typů v typovém systému s hierarchií

Váš úkolem je naprogramovat aplikaci, která načte definici jednoduchého typového systému a poté bude odpovídat na dotazy ohledně přiřaditelnosti mezi různými typy v tomto systému.

Pokud je program spuštěn s jiným počtem argumentů na příkazové řádce nežli 1, vypíše na standardní výstup text **Usage: Program.exe typesFile**. V opačném případě program pracuje ve dvou fázích, kdy první proběhne fáze zpracování deklarací typů, a pokud tato fáze skončí úspěšně, tak program dovolí uživateli zadávat dotazy.

## Formát vstupního souboru — Deklarace typů

Program čte jediný soubor zadaný na příkazové řádce. V tomto souboru se na každém řádku nachází deklarace jednoho typu. Deklarace může být dvojího typu:

**Jméno : Předek** Deklaruje typ *Jméno*, který dědí od typu *Předek*.  
**Jméno** Deklaruje typ *Jméno*, který dědí od typu **Object**

Aby mohl být typ deklarován, tak musí platit několik věcí:

- Typ ještě nesmí být deklarován. Jinak se vypíše chyba ***N*: Duplicate type *X*. First declared on line *M***, kde *N* je číslo řádku na kterém chyba nastala, *M* je číslo řádku kde byl typ prvně deklarován a *X* je jméno typu.
- Předek typu již musí být deklarován. Jinak se vypíše chyba ***N*: Non existing base type *X***, kde *N* je číslo řádku na kterém chyba nastala a *X* je jméno typu.
- Pro speciální typ **Object** se předpokládá, že je deklarovaný na nultém řádku vstupního programu a nemá přímo žádného předka (jakožto jediný typ).

Pokud během načítání vstupního souboru nastane některá z výše popsaných chyb, tak daný řádek nedeklaruje žádný nový typ, ale načítání pokračuje dál. Pokud se soubor dočte a nedošlo k žádné chybě, tak se přejde k další fázi. Jinak se vypíše ***K errors. Stopping.***, kde *K* je celkový počet vypsaných chyb.

## Formát dotazů

Program v interaktivním režimu čte ze standardního vstupu dotazy, na které hned odpovídá. Dotaz je jen jednoho typu a má následující formát:

$$A = B_n B_{n-1} \dots B_1$$

Kde  $n \geq 1$  a odpověď na tento dotaz je jedna z následujících:

- Assignment  $A = B_1$  is valid, pokud je možné přetypovat typ  $B_1$  na typ  $B_2, \dots$ , typ  $B_{n-1}$  na typ  $B_n$  a typ  $B_n$ , lze přiřadit do typu  $A$ .
- Assignment  $A = B_1$  is invalid, pokud výše popsané v nějakém kroku selže (přetypování nebo přiřazení není možné provést)
- Cannot find type  $X$ , pokud některý z typů  $A, B_1, \dots, B_n$  nebyl deklarovaný, kde  $X$  je název některého nedeklarovaného typu.

**Přiřazení** z typu  $Y$  do typu  $X$  (tedy  $X = Y$ ) lze provést, pokud  $X$  je přímo  $Y$  nebo  $X$  je nějakým předkem  $Y$  (do libovolné úrovně)

**Přetypování** typu  $X$  na typ  $Y$  (tedy  $Y X$ ) lze provést, pokud  $X$  je přímo  $Y$  nebo pokud  $Y$  lze přiřadit do  $X$  ( $X = Y$  je validní) nebo pokud  $X$  je nějakým potomkem  $Y$  (do libovolné úrovně).

### Garance

- Můžete předpokládat, že vstupní soubor je vždy ve správném formátu (syntaktickém), dle zadání. Může se ale stát, že dojde k logické chybě při pokusu o deklaraci typu.
- Můžete předpokládat, že dotazy jsou vždy zadávány ve formátu specifikovaném v zadání. Je však možné jednotlivé dotazy oddělovat prázdným řádkem, na který se neodpovídá.

### Příklady

Řádky začínající dolarem (\$) jsou ty, které reprezentují spuštění programu. Řádky se znaménkem větší než (>) jsou ty, které program vypisuje na standardní výstup. Řádky se znaménkem menší než (<) jsou ty, které uživatel píše na standardní vstup. Žádné z těchto znaků nejsou součástí výstupu programu, slouží jen pro lepší orientaci v příkladech. Testovací soubory a zadání v elektronické formě můžete stáhnout na adrese <http://d3s.mff.cuni.cz/~kliber/test03.zip>.

```
$ Program.exe
> Usage: Program.exe typesfile
```

---

\$ Program.exe data.invalid	\$ cat data.invalid
> 1: Duplicate type Object. First declared on line 0	> Object
> 2: Non existing base type A.	> A : A
> 3: Non existing base type A.	> B : A
> 6: Duplicate type D. First declared on line 5	> C
> 4 errors. Stopping.	> D
	> D : C

---

\$ Program.exe data	\$ cat data
< Object = Base	> Base
> Assignment Object = Base is valid	> Derived : Base
< Base = Derived	
> Assignment Base = Derived is valid	
< Object = Derived	
> Assignment Object = Derived is valid	
< Base = Object	
> Assignment Base = Object is invalid	
< Base = Base Object	
> Assignment Base = Object is valid	