

Během zkoušky aktivně komunikujte se zkoušejícími, nebojte se zeptat. Nad zadáním přemýšlejte, často je těžší zjistit *co* řešit než *jak*. Kód pište čistě a průběžně ho vylepšujte. Nesnažte se vyřešit všechny příklady naráz, hodnotit budeme především podle kvality vašich myšlenek a vašeho kódu.

V celém zadání neuvažujeme problematiku přesnosti čísel typu `double` danou jejich reprezentací. Pro porovnání používejte standardní operátor `==`.

1. Naimplementujte metody `getValue()` a `copy()` ve všech třídách. Metoda `copy()` vrací *hlubokou kopii* původního výrazu.
2. Naimplementujte metodu `derivative()` počítající derivace pro všechny typy výrazů kromě násobení (`Multiply`). U výrazu `Power` se omezte na to, že výraz exponentu neobsahuje závislou proměnnou (pro závislou proměnnou x jsou přípustné výrazy typu x^c , nikoliv c^x).
3. Přidejte podporu pro vzor *visitor* a naimplementujte třídu `PrintExpressionVisitor`. Použijte infixovou notaci. Pozor na závorky.
4. Doplněte podporu pro *iterator* (viz metoda `variableIterator()`), který projde výraz a vrátí postupně jména všech obsažených proměnných. Jména budou seřazena vzestupně podle abecedy a nebudou se opakovat.
5. Naimplementujte *visitor* (`SimplifyExpressionVisitor`), který bude schopen symbolicky zjednodušit výrazy podle následujících pravidel:
 - (a) Sečtete konstanty pro všechny potomky `Add`.
 - (b) Odstraňte konstantu/potomka `Add` rovného nule.
 - (c) Pokud je alespoň jeden potomek `Multiply` roven nule, nahraďte celý výraz nulou.
 - (d) Vynásobte všechny konstantní potomky `Multiply`.
 - (e) Odstraňte potomka `Multiply` rovného 1.
 - (f) Pro `Power` zjednodušte: $0^c \rightarrow 0$, $1^c \rightarrow 1$, $c^0 \rightarrow 1$ a $c^1 \rightarrow c$.
 - (g) Pokud jsou všechny parametry libovolné funkce konstantami, nahraďte ji výsledkem.

Operátor `instanceof` je přípustný. Minimalizujte opakování kódu.

6. Naimplementujte metodu `derivative()` pro násobení. Použijte rekuzi.

Tabulka derivací pro závislou proměnnou x :

Výraz	Derivace	Poznámka
<code>c</code>	<code>0</code>	konstanta
<code>sin(x)</code>	<code>cos(x)</code>	sinus
<code>cos(x)</code>	<code>-sin(x)</code>	kosinus
<code>x^c</code>	<code>cx^{c-1}</code>	mocnina
<code>f(x) + g(x)</code>	<code>f'(x) + g'(x)</code>	součet funkcí
<code>f(x)g(x)</code>	<code>f'(x)g(x) + f(x)g'(x)</code>	součin funkcí
<code>f(g(x))</code>	<code>f'(g(x))g'(x)</code>	složená funkce