

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.

1. Vytvořte objekty reprezentující následující kus kódu: $[a \mapsto a(2)]([b \mapsto 3])$.
2. Napište metodu `Type.matches(Type)`, která rozhodne, jestli dané dva stromy reprezentují stejný typ (tzn. mají stejný tvar a uzly stejných typů). Pro zjištění konkrétního typu uzlu použijte dvojitý dispatch.
3. Napište metodu `Expression.rename(String oldName, String newName)`, která vrátí kopii tohoto výrazu se všemi výskyty proměnné `oldName` přejmenovanými na `newName`. Během přejmenování postupujte podle následujících pravidel:
 - Pokud e je číslo, pak přejmenování proměnné a na b ve výrazu e je opět výraz e .
 - Pokud e je proměnná a , pak přejmenování proměnné a na b ve výrazu e je výraz b .
 - Pokud e je proměnná v odlišná od a , pak přejmenování proměnné a na b ve výrazu e je opět výraz e .
 - Pokud e je funkce $[x \mapsto y]$ a x je odlišné od a , pak přejmenování proměnné a na b ve výrazu e je výraz $[x \mapsto y']$, kde y' získáme přejmenováním a na b v y .
 - Pokud e je funkce $[a \mapsto y]$, pak přejmenování proměnné a na b ve výrazu e je opět výraz e .
 - Pokud e je volání $f(x)$, pak přejmenování proměnné a na b ve výrazu e je výraz $f'(x')$, kde f' získáme přejmenováním a na b v f a x' získáme přejmenováním a na b v x .
4. Do třídy `Function` si přidejte proměnnou `varType` reprezentující typ parametru této funkce. Naprogramujte metodu `Expression.getType(Map<String, Type>)`, která vrátí typ daného výrazu nebo `null`, pokud se uživatel pokouší volat funkci s parametrem nevhodného typu.
5. Upravte vaše řešení předchozího příkladu tak, aby používalo nulový (prázdný) objekt.
6. Zobecněte celý kód tak, aby v něm bylo možné používat konstanty libovolného typu, ne jen typu `Integer`. Adekvátně upravte typovou kontrolu, vztah podtyp/nadtyp při ní nezohledňujte (např. typy `Object` a `String` považujte za navzájem typově nekompatibilní).