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: $[a \mapsto [b \mapsto a]](2)$.

2. Dopište metodu `Function.apply(Expression)`, která vrátí výsledek volání dané funkce s daným parametrem (tzn. vrátí kopii jejího těla, ve které je každý výskyt řídící proměnné nahrazen výrazem v parametru).

3. Napište tovární metody, které si budou evidovat již existující (pod)výrazy a v případě potřeby je budou recyklovat. Upravte zbylý kód tak, aby využíval pouze tyto tovární metody (tzn. aby nevolal konstruktory tříd `Variable`, `Number`, `Function` a `Invocation`). Keš s již existujícími výrazy nedělejte statickou — vytvořte pro ni třídu `ExpressionContext` a upravte zbytek kódu tak, aby s touto třídou vhodně pracoval. Při volání metody `Function.apply` můžete předpokládat, že funkce i substituovaný parametr patří ke stejnému kontextu.

4. Napište metodu `Expression.simplify`, která vrátí o jeden krok jednodušší verzi celého výrazu (nebo `null`, pokud daný výraz zjednodušit nelze). Při zjednodušování postupujte podle následujících pravidel:

   - Pokud je e je číslo nebo proměnná, pak zjednodušit nejde.
   - Pokud y' je zjednodušení výrazu y, pak $[x \mapsto y']$ je zjednodušení výrazu $[x \mapsto y]$.
   - Pokud x' je zjednodušení výrazu x a y' je zjednodušení výrazu y, tak $x'(y)$ i $x(y')$ jsou zjednodušení výrazu $x(y)$. Navíc, pokud x je funkce $[v \mapsto w]$, tak i výraz w, ve kterém je každý výskyt proměnné v nahrazen výrazem y, je zjednodušením výrazu $x(y)$.

5. Napište metodu `Expression.iterator`, která vrátí iterátor postupně zjednodušující tento výraz. Zjednodušení generujte "na požádání". Metodu `remove` nechte neimplementovanou.

6. Všechny čtyři tovární metody napsané ve třetím příkladu dělají prakticky to samé: pokud v keši není vhodný objekt, vyrobí ho a uloží ho do ní. Slučte jejich kód do jedné zobecněné metody.