

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. Naimplementujte metodu `Main.convert`, která podstrom souborů s kořenem v instanci `java.io.File` zkonvertuje na strom instancí `Entry`. Existenci symlinků ignorujte.
2. Pomocí rekurze naimplementujte metodu `SymLink.resolve`. Tato metoda by měla vrátit soubor, adresář nebo symlink, na který daný symlink ukazuje (např. pokud je v atributu `linkedEntry` hodnota `../soubor`, vrátí soubor, adresář nebo symlink se jménem `soubor` umístěný v rodičovském adresáři). Pokud cílový soubor neexistuje, vraťte `null`. Tip: použijte metodu `String.split`.
3. Do třídy `Entry` dopište podporu pro návrhový vzor návštěvník. Dále napište návštěvníka `DuVisitor`, jehož metoda `du` spočítá a vrátí celkovou velikost daného podstromu (adresáře a symlinky mají nulovou velikost). Pozor: pokud instance `File` sdílí stejnou instanci `FileContents`, započítejte její velikost jen jednou.
4. K atributu `Entry.parentEntry` doplňte modifikátor `final` a napište stavitele, kteří vám umožní postavit imutabilní hierarchii ze tříd `Directory`, `File` a `SymLink`. Pravděpodobně budete muset změnit parametry konstruktorů.
5. Naimplementujte metodu `Directory.find`, která ze všech souborů v podstromu daného adresáře vrátí ty, k nimž vrátil predikát `predicate` hodnotu `true`.
6. Napište metodu `Main.buildFS`, která vrátí strom souborů vybudovaný z parametru `classes`: jednotlivé třídy odpovídají souborům, jejich balíčky zase adresářům.