

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. Ve zdrojovém kódu máte třídy, které reprezentují cifry nula a jedna. Typ `ImmutableList<Digit>` se dá chápat jako zápis nějakého čísla v binárním formátu. Reprezentujte číslo 147.
2. Napište tovární metodu `DigitUtils.convert(int)`, která zadané číslo zkonvertuje na instanci `ImmutableList<Digit>`. Dbejte na efektivnost vašeho kódu.
3. Napište metodu `DigitUtils.print(ImmutableList<Digit>)`, která zadané číslo zaserializuje do řetězce a vrátí. Vyhněte se použití podmínek.
4. Napište metodu `DigitUtils.inc(ImmutableList<Digit>)`, která k danému číslu přičte jedničku. Přičtení jedničky implementujte pomocí následujícího algoritmu: odzadu měňte jedničky na nuly tak dlouho, dokud nenarazíte na nulu — tu změňte na jedničku. Využijte dvojitý dispatch.
5. Třída `AdditionWithStates` reprezentuje součet dvou čísel. Po vytvoření si drží dva sčítance, při prvním volání metody `getValue` je sečte a zapamatuje si výsledek. Dopíšte její kód, využijte návrhový vzor stav.
6. Přepište konverzi `ImmutableList<Digit>` na řetězec tak, aby rozumně využívala metodu `OtherUtils.fold`.