

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. Instance implementující rozhraní `B` reprezentují hodnoty trojhodnotové logiky: instance třídy `T` reprezentují pravdu, instance třídy `F` nepravdu a instance třídy `M` reprezentují nejistotu (možná). Napište metodu `B.not` (negace nejistoty je nejistota).
2. Dopište metodu `B.or(B)`, vyhněte se použití podmínek (disjunkce čehokoliv a nejistoty je nejistota).
3. Dopište kód metody `BUtils.forall`, která vrátí instanci `T`, pokud ve vstupním seznamu jsou jen instance `T`, instanci `M`, pokud ve vstupním seznamu jsou jen instance `T` nebo `M` a instanci `F` jinak. Minimalizujte počet podmínek, šikovně využijte hlavní myšlenku návrhového vzoru stav.
4. Naimplementujte metodu `BUtils.forall2`, která se chová naprosto stejně jako `BUtils.forall`, ale vhodně využívá metodu `OtherUtils.fold`.
5. Naimplementujte třídy `And`, `Or` a `Not`, které vám dovolí reprezentovat výraz v ternární logice jako strom. Dopište do nich metodu, která vrátí pravdivostní hodnotu tohoto výrazu.
6. Do předchozí struktury doplňte podporu pro návrhový vzor návštěvník, napište návštěvníka, který vrátí kopii stromu, ve které jsou instance třídy `M` nahrazené instancemi `F`.