

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.

Při implementaci `java.util.Iterator` nemusíte implementovat nepovinnou metodu `remove`.

1. Naprogramujte adaptér z `java.util.Iterator<E>` do `AnotherIterator<E>`.
2. Napište iterátor, který dostane dva iterátory a spojí je do jednoho tak, že nejprve vrátí všechny prvky z prvního a poté všechny prvky z druhého iterátoru. Implementaci udělejte co nejvíce efektivní, použijte v ní návrhový vzor stav.
3. Do třídy `TreeNode` dopište metodu, která vrátí iterátor přes všechny hodnoty v listech daného (pod)stromu.
4. Dopíšte iterátor do třídy `LL`, minimalizujte počet použitých podmínek.
5. Napište statickou metodu `<E> E collapse(Iterator<E> i, Collapser<E> c)`, která mezi prvky iterátoru a_1, \dots, a_n vloží operaci \oplus reprezentovanou objektem `Collapser` a vrátí výsledek $a_1 \oplus a_2 \oplus \dots \oplus a_n$. `Collapser` si nejprve musíte vhodně nadefinovat.
6. Napište metodu `createIterator(Object o)`, která dostane objekt s metodami `E get(int index)` a `int size()` a vrátí jeho iterátor.