

DU 2

Pokyny:

- Deadline na odovzdanie je 8.11.2023, 23:59:00.
- Odovzdava sa zdrojovy kod v ZIP archive do Teamsov.
- Piste citatelny a strukturovany kod, za spagety bude malo bodov!
- Tiez dbajte o dodrziavanie aspon zakladnych konvencii jazyka Java.
- ... samozrejme ziadne opisovanie kodu z internetov, za to budu sankcie.
- Dokopy mozete ziskat 10 bodov.
- V pripade lubovolnych nejasnosti piste, volajte, kricte.

Cast A: Memoizacia (5 bodov)

V baliku 'memo' vytvorte rozhranie Function s jedinou metodom `BigInteger eval(BigInteger i)` reprezentujuce funkcie nad celymi cislami potencialne velmi velkeho rozsahu. Vytvorte triedu Decorator, ktora podla rovnomenného navrhového vzoru reprezentuje dekoratory pre Function. Takisto vytvore konkrétny dekorator Cache, ktorý volanie `eval(BigInteger i)` vypocita nasledovne:

1. Ak počas života instance dekoratora nebola metóda eval volaná s argumentom i, funkcia bude vypocítaná a jej výsledok bude v tejto instancii uchovaný vo vhodnej datovej strukture.
2. V opačnom prípade najdeme a vrátime uložený výsledok z predosleho volania.

Otestujte svoju implementáciu na vlastnej triede implementujúcej Function, napr. pomocou rekurzívneho počítania n-teho člena fibonacciho postupnosti, kde avšak do Cache vieme uložiť len prvé volanie funkcie.

Cast B: Kombinácie (5 bodov)

V baliku 'comb' naprogramujte triedu `Combinations<T>`, ktorá umožňuje generovať kombinácie bez opakovania prvkov zoznamu pomocou implementácie rozhrania `Iterator<List<T>>`. Trieda `Combinations<T>` musí obsahovať konštruktor `Combinations(List<T> elements, int k)`, kde 'elements' je zoznam, z ktorého chceme generovať všetky podmnožiny (reprezentované ako zoznamy) veľkosti k.

Příklad použitia:

```
List<Character> a = new ArrayList<>();
for (char c = 'a'; c <= 'd'; c++) {
    a.add(c);
}
Combinations<Character> comb = new Combinations<>(a, 2);
while (comb.hasNext()) {
    System.out.println(comb.next());
}
```

}

Vystup:

[a, b]

[a, c]

[a, d]

[b, c]

[b, d]

[c, d]