

L3 užduotis

1. Išnagrinėkite Lab3_MaisosLentelės projekte pateiktą programinį kodą.
2. **(7 balai)** Realizuokite nerealizuotus programinio kodo metodus ir juos ištestuokite (testus galite inicijuoti tiek per projekto vartotojo sąsają, tiek per konsolę).

Klasėse HashMap ir HashMapOa reikia realizuoti šiuos metodus:

- remove(K key);
- containsValue(Object value) - patikrina ar atvaizdyje egzistuoja vienas ar daugiau raktų metodo argumente nurodytai reikšmei;
- replace(K key, V oldValue, V newValue) - pakeičia atvaizdyje egzistuojančio rakto *key* reikšmę *oldValue* į naują reikšmę *newValue* ir grąžina true. Jei raktas neegzistuoja atvaizdyje, ar jo reikšmė neatitinka metodo argumente nurodytos senosios reikšmės, pakeitimas nevykdomas ir grąžinama false.

3. **(3 balai)** Atlikite greitaveikos testavimą (naudokite JMH) su viena metodų pora iš žemiau pateiktos lentelės:

Nr.	Metodas 1	Metodas 2
1	Class HashMap: remove()	Class HashMapOa: remove()
2	Class HashMap: contains()	Class HashMapOa: contains()
3	Class HashMap: put()	Class HashMapOa: put()
4	Class HashMap: get()	Class HashMapOa: get()
5	Class HashMap: containsValue()	Class HashMapOa: containsValue()
6	Class HashMap: remove()	Class java.util.HashMap <E>: remove()
7	Class HashMap: contains()	Class java.util.HashMap <E>: contains()
8	Class HashMap: containsValue()	Class java.util.HashMap <E>: containsValue()
9	Class HashMap: put()	Class java.util.HashMap <E>: put()
10	Class HashMap: get()	Class java.util.HashMap <E>: get()
11	Class HashMapOa: remove()	Class java.util.HashMap <E>: remove()
12	Class HashMapOa: contains()	Class java.util.HashMap <E>: contains()
13	Class HashMapOa: containsValue()	Class java.util.HashMap <E>: containsValue()
14	Class HashMapOa: put()	Class java.util.HashMap <E>: put()
15	Class HashMapOa: get()	Class java.util.HashMap <E>: get()

Pastaba: Konkretų variantą gausite iš laboratorinių darbų dėstytojo.

Laisva forma parenkite ataskaitą. Ataskaitoje būtinai turi būti:

- Aprašyti tiriamieji metodai.
- Pateiktas tiriamųjų metodų asimptotinis sudėtingumas.
- Aprašyta greitaveikos testavimo metodika (testavimo algoritmas).
- Aprašomi kompiuterio, su kuriuo buvo atlikti greitaveikos testai, pagrindiniai parametrai (procesoriaus charakteristikos, atminties kiekis ir pan.).
- Algoritmų/metodų vykdymo laiko priklausomybės nuo įvesties duomenų kiekio grafikas.
- Išvados. Išvadose laisva forma turėtų būti pateikti atsakymai į šiuos klausimus:
 - Ar eksperimentiškai nustatyta vykdymo laiko priklausomybė nuo įvesties dydžio atitinka teorinį algoritmo/metodo asimptotinį sudėtingumą. Jei ne, kodėl?
 - Kurie iš tirtųjų metodų vykdymo laiko požiūriu yra geresni. Kodėl?

LD3 gynimas

Gynimo metu dėstytojai gali studentų prašyti atlikti žemiau pateiktas užduotis ir atsakyti į žemiau pateiktus klausimus. Klausimų ir užduočių sąrašas nėra baigtinis - dėstytojai gynimo metu gali užduoti su tema susijusių klausimų ar praktinių užduočių, kurių nėra šiame sąrašė.

Galimos praktinės užduotys

Realizuokite vieną ar kelis metodus iš šio sąrašo:

Grąžinama reikšmė	Metodo aprašymas
java.util.Set<K>	keySet() Grąžina šio atvaizdžio raktų aibę.
java.util.List<V>	values() Grąžina visų atvaizdžio reikšmių sąrašą.
int	getNumberOfCollisions() Grąžina kolizijų skaičių.
int	averageChainSize() Grąžina vidutinį grandinės ilgį.
V	replace(K key, V value) Pakeičia atvaizdyje egzistuojantį raktą <i>key</i> atitinkančią reikšmę <i>value</i> ir grąžina senąją reikšmę. Jei raktas <i>key</i> neegzistuoja atvaizdyje, pora (<i>key</i> , <i>value</i>) talpinama atvaizdyje ir grąžinama null.
void	replaceAll(V oldValue, V newValue) Pakeičia visų atvaizdžio porų reikšmes nauja reikšme <i>newValue</i> , jei senoji reikšmė yra lygi <i>oldValue</i> .
K	putIfAbsent(K key, V value) Jei pora (<i>key</i> , <i>value</i>) neegzistuoja atvaizdyje, ji įrašoma ir grąžinama null. Kitu atveju grąžinama atvaizdyje jau egzistuojantį raktą <i>key</i> atitinkanti reikšmė

void	putAll(Map<K,V> map) Kopijuoja visas raktų-reikšmių poras iš map į šį atvaizdį.
------	--

Galimi teoriniai klausimai

1. Koks yra įvairių laboratoriniame darbe realizuotų operacijų asimptotinis sudėtingumas?
2. Kokie yra maišos lentelės kolizijų sprendimo atvirų grandinėlių metodu privalumai ir trūkumai?
3. Kokie yra maišos lentelės kolizijų sprendimo atviros adresacijos metodu privalumai ir trūkumai?
4. Mokėkite paaiškinti įvairius su programiniu kodu susijusius niuansus.