



Кафедра вычислительной техники
Основы профессиональной деятельности

Дополнительная лабораторная работа №0
Разные списки котов

Выполнила: Берелехис Светлана Михайловна

P3112

Санкт-Петербург

2021

Оглавление

Задание:	2
Пример работы 1 этапа:	6
Пример работы 2 этапа:	6
Пример работы 3 этапа:	7
Вопросы и ответы:	8
Вывод:	8

Задание:

Этап 1: Реализация модуля хранения на двух массивах

Храним соответствие <Имя кота> → <цвет, возраст, вес>

- Имя кота: String; служит идентификатором, id
- Цвет: String
- Возраст: Int
- Вес: Int

Чтобы хранить пару из ключа и значения, в одном массиве хранится ключ, в другом, по тому же индексу — значение.

Одновременно может храниться не более 1000 пар.

Для вхождения “Имя → цвет, возраст, вес” фиксируется формат вывода, на усмотрение исполнителя. Вывод вхождений осуществляется всегда в этом формате.

Команды для взаимодействия через CLI:

create

- Гарантируется уникальность id, т.е. в массиве ключей не хранится ключа равного id.

- Формат ввода: “create <имя> <цвет> <возраст> <вес>”

- Выводит созданное вхождение

read

- Гарантируется корректность id, т.е. в массиве ключей точно хранится ключ равный id.

- Формат ввода: “read <имя>”

- Выводит прочитанное вхождение

delete id

- Гарантируется корректность id, т.е. в массиве ключей точно хранится ключ равный id.

- Формат ввода: "delete <имя>"

- Выводит ОК

readall

- Выводит все пары ключей и значений в произвольном порядке

- Формат ввода: "readall"

- Каждое вхождение выводится с новой строки.

По одной команде на строчке.

этап 2: Реализация модуля хранения на связном списке из

пар

Храним соответствие <Имя кота> → <цвет, возраст, вес>

- Имя кота: String; служит идентификатором

- Цвет: String

- Возраст: Int

- Вес: Float

Чтобы хранить пару из ключа и значения, храним в связном списке пары из имени кота и data class'a, соответствующего значению <цвет, возраст, вес>

Одновременно может храниться любое количество пар.

Для вхождения "Имя → цвет, возраст, вес" фиксируется формат вывода, на усмотрение исполнителя. Вывод вхождений осуществляется всегда в этом формате.

Команды для взаимодействия через CLI:

create

- Не гарантируется уникальность id, т.е. в списке не обязательно хранится пара, в которой первый компонент равен id.

- Формат ввода: "create <имя> <цвет> <возраст> <вес>"

- Выводит созданное вхождение или сообщение об ошибке "Create: already exists"

read

- Не гарантируется корректность id, т.е. в списке не обязательно хранится пара, в которой первый компонент равен id.

- Формат ввода: "read <имя>"

- Выводит прочитанное вхождение или сообщение об ошибке “Read: not found”

delete id

- Не гарантируется корректность id, т.е. в списке не обязательно хранится пара, в которой первый компонент равен id.

- Формат ввода: “delete <имя>”

- Выводит “OK” или сообщение об ошибке “Delete: not found”

readall

- Выводит все пары ключей и значений в произвольном порядке

- Формат ввода: “readall”

- Каждое вхождение выводится с новой строки.

По одной команде на строчке.

этап 3: Реализация модуля хранения на самописной хэш-таблице

Храним соответствие <Имя кота> → <цвет, возраст, вес>

- Имя кота: String; служит идентификатором
- Цвет: String
- Возраст: Int
- Вес: Float

Чтобы хранить пару из ключа и значения, храним в таблице пары из имени кота и data class’a, соответствующего значению <цвет, возраст, вес>

Хэш-таблица реализована как массив со связными списками пар.

Одновременно может храниться любое количество пар.

Алгоритм хэширования — на выбор студента.

Для вхождения “Имя → цвет, возраст, вес” фиксируется формат вывода, на усмотрение исполнителя. Вывод вхождений осуществляется всегда в этом формате.

Команды для взаимодействия через CLI:

create

- Не гарантируется уникальность id, т.е. в списке не обязательно хранится пара, в которой первый компонент равен id.

- Формат ввода: “create <имя> <цвет> <возраст> <вес>”

- Выводит созданное вхождение

read

- Не гарантируется корректность id, т.е. в списке не обязательно хранится пара,

в которой первый компонент равен id.

- Формат ввода: "read <имя>"
- Выводит прочитанное вхождение

update id

- Не гарантируется корректность id, т.е. в списке не обязательно хранится пара, в которой первый компонент равен id.

- Формат ввода: "update <имя> <цвет> <возраст> <вес>"
- Выводит прочитанное вхождение

delete id

- Не гарантируется корректность id, т.е. в списке не обязательно хранится пара, в которой первый компонент равен id.

- Формат ввода: "delete <имя>"
- Выводит ОК
- where weight=<value>
- Выводит все вхождения где вес кота равен value с epsilon = 0.001
- Формат ввода: " where weight=<value>"

readall

- Выводит все пары ключей и значений в произвольном порядке
- Формат ввода: "readall"
- Каждое вхождение выводится с новой строки

задача - реализовать лабораторную работу на языке kotlin, залить ее на гит, что бы в репозитории было 3 ветки соответствующие каждому этапу лабораторной работе

написать отчет - в котором приложить примеры работы каждого этапа, написать развернутый вывод с подробным указанием, что было сложно и просто и тд,

а так же в отчет включить все уточняющие вопросы, что были заданы к работе.

ВАЖНО

НЕ использовать классы

можно использовать ТОЛЬКО методы и DataClass

Пример работы 1 этапа:

Ввод	Вывод
create Tom white 12 3	Tom->Cat(color=white, age=12, weight=3)
delete Tom	OK
readall	
create Tom black 12 5	Tom->Cat(color=black, age=12, weight=5)
readall	Tom->Cat(color=black, age=12, weight=5)
create Rick yellow 3 10	Rick->Cat(color=yellow, age=3, weight=10)
readall	Tom->Cat(color=black, age=12, weight=5) Rick->Cat(color=yellow, age=3, weight=10)
create Margo black 15 7	Margo->Cat(color=black, age=15, weight=7)
readall	Tom->Cat(color=black, age=12, weight=5) Rick->Cat(color=yellow, age=3, weight=10) Margo->Cat(color=black, age=15, weight=7)
read Margo	Margo->Cat(color=black, age=15, weight=7)

Пример работы 2 этапа:

Ввод	Вывод
create Tom white 12 3	Tom->Catt(color=white, age=12, weight=3.0)
create Tom black 32 33	Create: already exists
delete Tom	Tom->Catt(color=white, age=12, weight=3.0)
delete Tom	Delete: not found
create Tom black 12 5	Tom->Catt(color=black, age=12, weight=5.0)
readall	Tom->Catt(color=black, age=12, weight=5.0)
read Tom	Tom->Catt(color=black, age=12, weight=5.0)
create Rick yellow 3 10	Rick->Catt(color=yellow, age=3, weight=10.0)
read Margo	Read: not found
readall	Tom->Catt(color=black, age=12, weight=5.0) Rick->Catt(color=yellow, age=3, weight=10.0)
create Margo black 15 7	Margo->Catt(color=black, age=15, weight=7.0)
readall	Tom->Catt(color=black, age=12, weight=5.0) Rick->Catt(color=yellow, age=3, weight=10.0) Margo->Catt(color=black, age=15, weight=7.0)

read Margo	Margo->Catt(color=black, age=15, weight=7.0)
delete Rick	Rick->Catt(color=yellow, age=3, weight=10.0)
readall	Tom->Catt(color=black, age=12, weight=5.0) Margo->Catt(color=black, age=15, weight=7.0)
read Margo	Margo->Catt(color=black, age=15, weight=7.0)

Пример работы 3 этапа:

Ввод	Вывод
create Tom white 12 13	Tom->CatH(color=white, age=12, weight=13.0)
create Tom black 32 33	Create: already exists
delete Tom	Tom->CatH(color=white, age=12, weight=13.0)
delete Tom	Delete: not found
create Tom black 12 13.0001	Tom->CatH(color=black, age=12, weight=13.0001)
readall	Tom->CatH(color=black, age=12, weight=13.0001)
read Tom	Tom->CatH(color=black, age=12, weight=13.0001)
create Rick yellow 33 12.9999	Rick->CatH(color=yellow, age=33, weight=12.9999)
where weight=13	Tom->CatH(color=black, age=12, weight=13.0001) Rick->CatH(color=yellow, age=33, weight=12.9999)
update Tom red 22 33	Tom->CatH(color=red, age=22, weight=33.0)
update Margo red 22 33	Update: not found
read Margo	Read: not found
readall	Tom->CatH(color=red, age=22, weight=33.0) Rick->CatH(color=yellow, age=33, weight=12.9999)
create Margo black 15 7	Margo->CatH(color=black, age=15, weight=7.0)
readall	Tom->CatH(color=red, age=22, weight=33.0) Margo->CatH(color=black, age=15, weight=7.0)

	Rick->CatH(color=yellow, age=33, weight=12.9999)
read Margo	Margo->CatH(color=black, age=15, weight=7.0)
delete Rick	Rick->CatH(color=yellow, age=33, weight=12.9999)
readall	Tom->CatH(color=red, age=22, weight=33.0) Margo->CatH(color=black, age=15, weight=7.0)
read Margo	Margo->CatH(color=black, age=15, weight=7.0)

Вопросы и ответы:

Вопрос	Ответ
Ограничений по времени нет или совсем в лоб делать не надо?	Нет ограничений
Вопрос про реализацию на хеш-таблице. Есть ли ограничение на количество котов?	нет
правильно будет использовать связный список для разрешения проблемы коллизий? (3 этап)	да
вопрос по 3 этапу: с некорректными id делать то же, что и во 2?	+

Вывод:

Для меня основной сложностью был синтаксис kotlin, потому что я трогала этот язык в первый раз. Если бы я писала подобную программу на Java, я бы использовала классы, поэтому в том, что получилось (особенно в 3 этапе) нарушены некоторые принципы SOLID, что печально.

При последовательном выполнении этапов работы, сложностей с алгоритмической частью не возникло.