Задания по курсу объектно-ориентированного программирования (ООП)

Для выполнения заданий необходимо создать аккаунт на публичном интернет ресурсе github.com и репозиторий «ООР».

Репозиторий должен быть доступен для чтения лектору(alexander-a-vlasov) и преподавателю ведущему семинарские занятия.

Задания выполняются на языке Java. Рекомендуется использовать IDE IntelliJ IDEA версии Community.

Все решения должны сопровождаться набором тестов проверяющего корректность выполненного задания.

Для выполнения первого задания «Пирамидальная сортировка» необходимо создать проект Task_1_1. Следующие задания именовать соответствующим образом.

Зеленым цветом обозначены примеры к заданиям, которые можно использовать в качестве первого теста.

Серым цветом обозначены дополнительные требования к задаче, которые не обязательны к реализации, но приносят дополнительные баллы

Желтым цветом отмечены комментарии к задаче. Дополнительные условия и ограничения.

1. Знакомство с языком программирования Java

1.1. Пирамидальная сортировка

Реализовать классический алгоритм пирамидальной сортировки с набором тестов.

вход:
{5,4,3,2,1}
выход:
{1,2,3,4,5}

1.2. Определить наличие подстроки в файле

На вход подаётся имя файла и строка, которую надо найти. Строка может содержать буквы любого алфавита в кодировке UTF-8.

Реализовать функцию определяющую индекс начала каждого вхождения заданной подстроки.

Размер входного файла может быть на много больше объема оперативной памяти вычислительного устройства. Размер искомой подстроки во много раз меньше доступного объема оперативной памяти.

Файл input.txt : Я хочу пирог!
вход:
input.txt
пирог
выход:
{7}

Файл input.txt : Я хочу сок!
вход:
input.txt
пирог
выход:
{}

2. Контейнеры

2.1. Стек (stack)

Реализовать класс стек с поддержкой операций push (добавить элемент), pop(взять элемент), и count(узнать количество элементов). Элементы стека могут быть разных типов. Должна быть возможность использовать стандартный итератор для обхода контейнера.

В реализации запрещено использовать стандартные контейнеры языка Java, кроме массива.

вход:
push(2)
push(7)
pop
count
выход:
{2}

2.2. Очередь с приоритетами (priority queue)

Реализовать класс очередь с приоритетами. Поддерживается две операции Insert(ключ, значение) — вставка в очередь и extrack_minimum(ключ, значение) — удаляет и возвращает значение с максимальным ключом. Тип данных ключа и значения могут быть произвольными. Должны поддерживаться стандартные механизмы итерирования контейнера и Stream API.

вход:
Insert 200, «собака»
Insert 10, «человек»
ExtractMax –> 200, «собака»
Insert 5, «пингвин»
Insert 500, «попугай»
ExtractMax -> 500, «попугай»
выход:
{{5,«пингвин»},{10,«человек»}}

2.3. Дерево квадрантов (quadtree)

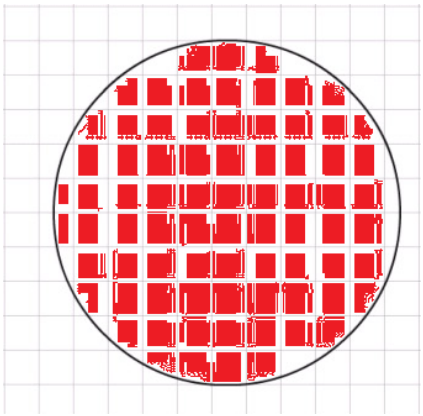
Для множества начальных объектов на плоскости построить «Quadtree»(Дерево квадрантов) со следующими операциями:

1) добавить новый объект;
2) удалить существующий объект;
3) Получить объект соответствующий координатам (x,y), значение координат произвольные;
4) получения все объекты в заданном прямоугольнике;

Каждому узлу дерева соответствует заданный пользователем объект.

Должны поддерживаться стандартные механизмы итерирования контейнера и исключительная ситуация ConcurrentModificationException.

На основе созданного контейнера реализовать функцию построения круга с заданным центром, радиусом, погрешностью определения принадлежности точки кругу. Созданный контейнер должен содержать минимальное количество узлов.



Пример круга с не минимальным количеством узлов.

вход:
Quadtree t=CreateCircle(xcenter=3,ycenter=4, r=5, tolerance=0.01)
print t.get(x=0,y=0)
выход:
“круг”

3. Типы данных

3.1. Григорианский календарь

Реализовать класс григорианского календаря со следующими особенностями поведения:

1) каждый четвертый год високосный (в високосный год добавляется дата 29 февраля)
2) каждый завершающий год века, если он не делится на 400, не является високосным
3) для заданной даты определить день недели.
4) Поддержка арифметических операций сложение и вычитание (дата – дата; дата ± день, месяц, год)

Выполнить операции с календарем:

1) Какой день недели будет через 1024 дня?
2) Сколько лет, месяцев и дней назад был день победы 9 мая 1945 года?
3) В какой день недели вы родились?
4) Какой месяц будет через 17 недель?
5) Сколько дней до нового года?
6) Ближайшая пятница 13-го числа месяца?

Использование классов java.util.Date, java.util.Calendar, пакета java.time и других сторонних библиотек работы с датами допустимо только в тестах.

3.2. Зачетная книжка

Реализовать класс электронная зачетной книжки студента ФИТ и обеспечить следующие функции:

1) текущий средний бал за все время обучения;
2) может ли студент получить «красный» диплом с отличием;
3) будет ли повышенная стипендия в этом семестре.

Для первого теста используйте данные из Вашей зачётной книжки

Требования для диплома с отличием:

- 75% оценок в приложении к диплому(последняя оценка) – «отлично»
- Нет в зачетной книжке итоговых оценок «удовлетворительно»
- Квалификационная работа защищена на «отлично»

3.3. Упорядоченные множества

Реализовать библиотеку для выполнения классических операций над объектами: множество с , частично-упорядоченное множество(ЧУМ), решёточно-упорядоченное множество(РУМ), линейно-упорядоченное множество(ЛУМ):

1) проверка выполнения транзитивность для операции сравнения между элементами массива;
2) топологическая сортировка массива по возрастанию (текущий элемент отсортированного массива больше или не сравним чем все предыдущие);
3) поиск максимальных элементов в массиве.

С помощью созданной библиотеки решите прикладную задачу «Найти самых умных студентов в группе».

вход:
Группа: Мария, Василий, Татьяна, Дмитрий
1) Мария > Василия
2) Вероника > Василия
3) Василий > Татьяны
Выход:
Мария, Вероника, Дмитрий

4. Консольные приложения

4.1. Калькулятор

Реализовать калькулятор инженера для вещественных чисел. Пользователь вводит на стандартный поток ввода выражение в префиксном виде. Калькулятор вычисляет значение и выводит на стандартный поток вывода. Кроме стандартных операций (+, -, *, /) есть набор функций (log, pow, sqrt, sin, cos).

С помощью динамической загрузки библиотеки(plugin) переопределить поведение базовых операций. Добавить поддержку работы с градусами и комплексными числами.

вход:
sin + - 1 2 1
выход:
0

4.2. Записная книжка

Сделать записную книжку с набором функций доступных с командной строки:

- Добавить запись
- Удалить запись
- Вывести все записи, отсортированные по времени добавления
- Вывести отсортированные по времени добавления записи из заданного интервала времени, содержащие в заголовке ключевые слова.

Данные записной книги сериализуются в файл формата JSON. Для работы с форматом JSON рекомендуется использовать библиотеки Jackson или Gson. Для анализа параметров коммандной сроки, так же рекомендуется использовать сторонние библиотеки.

Примеры:
notebook -add “Моя заметка” “Очень важная заметка”
notebook -rm “Моя заметка”
notebook -show
notebook -show “14.12.2019 7:00” “17.12.2019 13:00” “МоЯ” “Твоя” “мне”