Лабораторная 2

Классы

Задание:

«Перевести» на классы свой вариант из Лабораторной #1.

На вход теперь будет подаваться массив классов, выходные данные и их формат остаются теми же. Классы создавать на свое усмотрение, но в конце файла есть пара советов по их созданию и делению. Учесть, чтоб входные классы могли меняться, при этом не влияя на остальной код.

Входные данные: созданные объекты в соотвествии с условием.

Варианты заданий так же переписаны с учетом задания на классы, сами задания при этом не изменились. Результаты должны остаться с первой лабораторной. Изменен формат входных данных.

После «перевода» на классы — показать. Будут добавлены какие-то еще свойства ко входным данным, возможно новый их тип. Иными словами, лабораторная состоит из двух частей: перевод предыдущей на классы и изменение под меняющиеся условия «заказчика». Рассчитана на 2 пары.

Варианты заданий:

1. Покупки. На вход поступает список покупок: идентификатор покупки, количество, цена единицы товара. Необходимо вывести их в формате id - цена - количество - сумма. Также необходимо вычислить показатели: самый дорогой товар(за единицу), самый дешевый товар(за единицу), наибольшее количество, наименьшее количество, самая дорогая покупка, самая дешевая покупка, сумма покупок, сумма количества.

Входные данные:

Покупка 1 - Количество 1, Цена: 15;

Покупка 2 - Количество 2.3, Цена: 13.3

Покупка 3 - Количество 4.0, Цена: 18.1

Покупка 4 - Количество 0.1, Цена: 11.0

Покупка 5 - Количество 5.5, Цена: 67.0

Покупка 6 - Количество 10.0, Цена: 23.0

Покупка 7 - Количество 213.0, Цена: 11.1

Покупка 8 - Количество 2.0, Цена: 18.1

Покупка 9 - Количество 3.4, Цена: 12.1

Покупка 10 - Количество 0.1, Цена: 23.3

Формат выходных данных:

Товар 1: Количество: 1.0, Цена: 15.0, Стоимость: 15.0

Товар 2: Количество: 2.3, Цена: 13.3, Стоимость: 30.59

//И ТД.

Самый дорогой товар: 7.

//И другие показатели

1. Расчет ресурсов. На вход поступает массив операций, содержащий расход на операцию и её идентификатор. Необходимо посчитать суммарный расход ресурсов. Также необходимо посчитать порядковый номер операции, после которой при выполнении следующей суммарный расход превысит 160 единиц. Как только такой расход достигается — идет новый отчет для следующей отметки. Легче показать на примере:

Входные данные:

Операция 1: расход ресурса 20;

Операция 2: расход ресурса 20;

Операция 3: расход ресурса 60;

Операция 4: расход ресурса 40;

Операция 5: расход ресурса 40;

Операция 6: расход ресурса 20;

Операция 7: расход ресурса 40;

Операция 8: расход ресурса 60;

Операция 9: расход ресурса 40;

Операция 10: расход ресурса 20;

Операция 11: расход ресурса 60;

Операция 12: расход ресурса 20;

Операция 13: расход ресурса 20;

Выходные данные:

Расход ресурса: 460

Отметки: 4, 8, 13

P.S. Я надеюсь, что посчитал правильно, но если будет что-то непонятно — спрашивайте.

1. Трудозатраты. Предыстория: в штабе имеется несколько разработчиков, которые в неделю могут взять задач на N-часов. На вход подается два массива. Первый содержит задачи, включая идентификатор и трудозатраты, второй — объекты классов с идентификаторами разработчиков и их максимальную занятость. Необходимо распределить задачи по разработчикам, чтобы суммарное количество его задач не превышало максимальное количество занятости. Также необходимо выделить задачу с самыми большим затратами и задачу с самыми маленьким затратами. Оптимизация значения не имеет, но нужно, чтобы все задачи были распределены между разработчиками.

Входные данные:

Задача 5: трудозатраты 11

Задача 12: трудозатраты 2

Задача 23: трудозатраты 5

Задача 11: трудозатраты 8

Задача 73: трудозатраты 23

Разработчик 1: максимальная занятость 15

Разработчик 2: максимальная занятость 30

Разработчик 4: максимальная занятость 15

Выходные данные:

Разработчик 1: 12, 23, 11

Разработчик 2: 73

Разработчик 4: 5

Задача с наибольшими трудозатратами: 73

Задача с наименьшими трудозатратами: 12

1. Бухгалтерия. Лупа и Пупа пришли получать зарплату, но в бухгалтерии все перепутали. Так как Лупа и Пупа работали сдельно, им выдали список выполненных задач с оплатой за них, в котором были не только они, но и куча других работников. Да и зарплату пришли получать не только они. На вход приходят массивы. Входной массив — выполненные задачи, их оплата и выполнивший работник. Необходимо высчитать зарплату (сумму оплаты) каждого из работников, а так же вывести самого высокооплачиваемого работника и самого низкооплачиваемого, среднюю оплату работников за задачу.

Входные данные:

Задача 1: оплата: 4500, выполнивший работник: 0

Задача 2: оплата: 17400, выполнивший работник: 3

Задача 3: оплата: 1380, выполнивший работник: 1

Задача 4: оплата: 1600, выполнивший работник: 2

Задача 5: оплата: 2500, выполнивший работник: 2

Задача 6: оплата: 13300, выполнивший работник: 0

Выходные данные:

Работник 0: 17800

Работник 1: 1380

Работник 2: 4100

Работник 3: 17400

Самый высокооплаичваемый: 0

Самый никзооплачиваемый: 1

1. Один студент решил крупно потратится на одну из Free-To-Play игр. Ему необходима сумма в M-внутриигровой валюты, когда у него сейчас только N. На вход подаются числа M и N. На вход подается массив операций, содержащий номер операции, полученную сумму игровой валюты и расход реальной валюты. Так же количество внутриигровой валюты за первый донат в каждом из типов умножается на 2 при той же реальной стоимости. Необходимо посчитать: суммарное количество внутриигровой валюты, полученной за эту цепочку, расход реальной валюты, итоговое количество внутриигровой валюты и перешагнул ли он свою необходимость. Также нужно посчитать, сколько раз он пополнял на каждую из сумм и сколько у него осталось почек после этого

Входные данные:

M: 3154

N: 38000

Операция 1: получено валюты 60, расход реальной валюты: 99

Операция 2: получено валюты 300, расход реальной валюты: 449

Операция 3: получено валюты 6480, расход реальной валюты: 9490

Операция 4: получено валюты 6480, расход реальной валюты: 9490

Операция 5: получено валюты 6480, расход реальной валюты: 9490

Операция 6: получено валюты 3280, расход реальной валюты: 4690

Операция 7: получено валюты 980, расход реальной валюты: 1390

Операция 8: получено валюты 300, расход реальной валюты: 449

Операция 9: получено валюты 300, расход реальной валюты: 449

Операция 10: получено валюты 60, расход реальной валюты: 99

Операция 11: получено валюты 1980, расход реальной валюты: 2790

Выходные данные:

Полученное количество внутриигровой валюты: 39100

Потраченная реальная валюта: 40785

Итого внутриигровой валюты: 42254

Перешагнул ли порог: Да

Покупка типа 0: 2

Покупка типа 1: 3

Покупка типа 2: 1

Покупка типа 3: 0

Покупка типа 4: 2

Покупка типа 5: 3

1. Имеется «обменник ресурсов». Всего имеется 4 «ранга» ресурсов, где каждый следующий ранг стоит 3 единиц ресурса предыдущего ранга. Так, имея 15 единиц ресурса ранга 1, можно получить 5 единиц ресурса ранга 2, 1 единицу ресурса ранга 3 и 0 единиц ресурса ранга 4. Так же за обмен на каждую полученную единицу тратится валюта. На вход поступает массив объектов класса, содержащий идентификатор ресурса и его начальное количество. Необходимо рассчитать итоговое количество ресурса каждого вида и суммарное затрачиваемое количество валюты на обмен. Также необходимо вывести самую дорогую и самую дешевую обменную операцию.

Входные данные:

Начальное количество ресурсов вида 1: 121

Начальное количество ресурсов вида 2: 49

Начальное количество ресурсов вида 3: 304

Выходные данные:

40->13->4 = 40\*20 + 13 \* 30 + 4 \* 50 = 1390

16->5->1 = 16\*20 + 5\*30 + 1\*50 = 520

101->33->11 = 101 \* 20 + 33\*30 + 11 \* 50 = 3560

Ресурсы вида 1: 1/1/1/4

Ресурсы вида 2: 1/1/2/1

Ресурсы вида 3: 1/2/0/11

Затраченная сумма валюты: 5470

Самая дорогая операция: ресурс 3, переход 1->2

Самая дешевая опреация: ресурс 2, переход 3->4

1. Быстродействие системы. На вход подается массив объекта класса операций, содержащий их время выполнения и стоимость выполнения. Необходимо найти суммарное время и стоимость выполнения всех операций, представить цепочки времени выполнения и стоимости, а так же посчитать время и стоимость на каждый вид операции.

Входные данные:

Операция 1: время выполнения 10, стоимость выполнения: 100

Операция 2: время выполнения 30, стоимость выполнения: 300

Операция 3: время выполнения 25, стоимость выполнения: 400

Операция 4: время выполнения 15, стоимость выполнения: 100

Операция 5: время выполнения 15, стоимость выполнения: 100

Операция 6: время выполнения 10, стоимость выполнения: 100

Операция 7: время выполнения 5, стоимость выполнения: 20

Операция 8: время выполнения 15, стоимость выполнения: 100

Операция 9: время выполнения 5, стоимость выполнения: 20

Выходные данные:

Суммарное время: 130

Суммарные затраты: 1240

Цепочка времени: { 10, 30, 25, 15, 15, 10, 5, 15, 5 }

Цепочка затрат: { 100, 300, 400, 100, 100, 100, 20, 100, 20 }

Операция 0: время - 20, стоимость - 200

Операция 1: время - 10, стоимость - 40

Операция 2: время - 45, стоимость - 300

Операция 3: время - 30, стоимость - 300

Операция 4: время - 25, стоимость - 400

1. Рейтинг фильмов. На вход подается массив объектов класса просмотренных фильмов, содержащий идентификатор фильма, оценку и жанр. Необходимо посчитать средний рейтинг фильмов, вывести фильм с максимальной и минимальной оценкой, а так же статистику по жанрам: сколько в каком жанре просмотрено фильмов и какая средняя оценка по жанрам.

Входные данные:

Фильм 1: оценка 5, жанр 1

Фильм 5: оценка 3, жанр 1

Фильм 17: оценка 4, жанр 2

Фильм 31: оценка 3, жанр 4

Выходные данные:

Средняя оценка: 3,75

Максимальная оценка: 5

Минималная оценка: 31

Жанр 1: просмотренно 2, средняя оценка 4

Жанр 2: просмотрено 1, средняя оценка 4

Жанр 4: прсмотрено 1, средняя оценка 3

1. Комплектующие. На вход подаются массив объектов класса комлпектующие, содержащий их тип, цену и мощность. Нужно рассчитать среднюю стоимость и мощность по всем видам комплектующих, максимальную а так же среднюю стоимость и мощность по каждому из видов. Входные данные:

Комплектующая 1: тип 1, цена 10, мощность 30

Комплектующая 2: тип 2, цена 15, мощность 40

Комплектующая 3: тип 1, цена 20, мощность 20

Комплектующая 4: тип 4, цена 25, мощность 15

Комплектующая 5: тип 3, цена 20, мощность 20

Комплектующая 6: тип 3, цена 15, мощность 40

Комплектующая 7: тип 3, цена 30, мощность 50

Выходные данные:

Средняя стоимость: 19.28

Средняя мощность: 30.71

Максимальная стоимость: 30

Максимальная мощность: 40

Комплектующая типа 1: средняя стоимость: 17.5; средняя мощность: 25

Комплектующая типа 2: средняя стоимость: 15; средняя мощность: 40

Комплектуюшая типа 3: средняя стоимость: 21.6; средняя мощность: 36.6

Комплектующая типа 4: средняя стоимость: 25; средняя мощность: 15

1. Кофемашина. На вход подается массив объектов класа кофе, содержащий стоимость кофе, себестоимость воды, себестоимость кофейных зерен, количество заказов. Нужно вычислить самый прибыльный кофе, самый малоприбыльный кофе, среднюю прибыль, а так же прибыль с единицы каждого кофе и со всей партии.

Входные данные:

Кофе 1: стоимость 20, себестоимость воды 2, себестоимость зерен 10, количество заказов 5

Кофе 2: стоимость 15, себестоимость воды 7, себестоимость зерен 5, количество заказов 10

Кофе 3: стоимость 10, себестоимость воды 5, себестоимость зерен 0, количество заказов 5

Выходные данные:

Самое прибильное кофе: кофе 0

Самое малоприбыльное кофе: кофе 1

Средняя прибыль: 5

Кофе 0: прибыль с единицы - 8, прибыль с партии - 40.

Кофе 1: прибыль с единицы - 3, прибыль с партии - 30.

Кофе 2: прибыль с единицы - 5, прибыль с партии - 25.

1. Вовремя ли пришел студент. На вход подаются массив объектов класса приходов, содержащий идентификатор студента и его время опоздания. Необходимо вывести количество опозданий каждого студента, его среднее время опоздания и общее количество опозданий из всех приходов.

Входные данные:

Приход 1: идентификатор студента 1, время опоздания: -1

Приход 2: идентификатор студента 2, время опоздания: -2

Приход 3: идентификатор студента 7, время опоздания: 5

Приход 4: идентификатор студента 4, время опоздания: 4

Приход 5: идентификатор студента 7, время опоздания: 3

Приход 6: идентификатор студента 2, время опоздания: -2

Приход 7: идентификатор студента 1, время опоздания: -1

Приход 8: идентификатор студента 4, время опоздания: 2

Приход 9: идентификатор студента 2, время опоздания: 0

Выходные данные:

Общее количество опозданий:4

Студент 1: опозданий - 0, среднее время опоздания - 0

Студент 2: опозданий - 0, среднее время опоздания - 0

Студент 4: опозданий - 2, среднее время опоздания - 2

Студент 7: опозданий - 2, среднее время опозданий - 4

1. Распил. Существует 4 участка заготовки древесины, где с каждой территории за заход можно добыть определенное количество древесины и стоимость. На вход подается массив объектов класса единицы распила, содержащий цену распила и добываемую древесину. Необходимо вычислить среднее количество добываемой древесины по всей цепочке, среднюю стоимость распила по всей цепочке, а так же суммарное количество древесины и цены. Также вывести статистику по территориям.

Входные данные:

Добыча 1: цена распила 10, добываемая древесина 20

Добыча 2: цена распила 15, добываемая древесина 10

Добыча 3: цена распила 5, добываемая древесина 30

Добыча 4: цена распила 10, добываемая древесина 20

Добыча 5: цена распила 10, добываемая древесина 20

Добыча 6: цена распила 15, добываемая древесина 10

Добыча 7: цена распила 15, добываемая древесина 10

Добыча 8: цена распила 10, добываемая древесина 20

Добыча 9: цена распила 5, добываемая древесина 30

Выходные данные:

Среднее количество добываемой древесины: 18.888888

Средняя цена: 10.55555555

Суммарное количество древесины: 170

Суммарная цена за распил: 95

Территория 0: количество древесины - 80, цена добычи - 120

Территория 1: количество древесины - 30, цена добычи - 45

Территория 2: количество древесины - 60, цена добычи - 10

1. Подсчеты. На вход подаются массив объектов класса изменений с полями идентификатора величины изменяемой величины, значения, на которую величина изменятся и тип её изменения, 0 в случае добавления простого числа, 1 — в случае, если изменение идет на процент. Необходимо рассчитать конечные значения всех величин, найти максимальную прибавку (в числе). Начальное значение каждой величины - 100.

Входные данные:

Изменение 1: величина 1, значение изменения 5, тип изменения 1

Изменение 2: величина 3, значение изменения 30, тип изменения 0

Изменение 3: величина 4, значение изменения 20, тип изменения 0

Изменение 4: величина 2, значение изменения 10, тип изменения 1

Изменение 5: величина 2, значение изменения 5, тип изменения 1

Изменение 6: величина 3, значение изменения 10, тип изменения 0

Изменение 7: величина 4, значение изменения 35, тип изменения 1

Изменение 8: величина 1, значение изменения 45, тип изменения 1

Изменение 9: величина 1, значение изменения 10, тип изменения 0

Выходные данные:

Величина 1: 165, максимальная прибавка - 45

Величина 2: 115.5, максимальная прибавка - 10

Величина 3: 143, максимальная прибавка - 30

Величина 4: 175.5, максимальная прибавка - 45.5

Best Practices по созданию классов:

1. С помощью интерфейсов можно отвязать конкретную сущность от конкретной реализации, оставив лишь нужные свойства. Например, если вам нужно посчитать условную калорийность всех компонентов продукта, вам не нужно знать, где этот продукт был сделан. Лишь калорийность.
2. Каждый класс пишется в отдельном файле. Можно дробить на пакеты.
3. Подумайте над тем, чтоб вынести повторяющиеся данные в отдельный класс, который будет полем у класса, где эти данные повторялись.
4. Классы условно можно поделить на два типа: модели, содержащие только свойства, несущие только информацию, в логике которых только их изменения и выдача и «исполняемые», где в методах заключена логика обработки модели.
5. В Java принято моделям, классам, которые только хранят информацию делать все поля с модификатором доступа private, после чего обращаться к ним через getter-ы/setter-ы. Что-то похожее есть и в C#. Такой подход позволяет контролировать все запросы значений полей или их изменения.  
   Пример:

public class Model {

private String someField;

public String getSomeField() {

//Можем как-то обработать эту выдачу

return someField;

}

public void setSomeField(String value) {

//Можем сделать какие-то проверки на значение

this.someField = value;

}

}

Model model = new Model();

model.setSomeField("Значение");

System.out.println(model.getSomeField());