

Домашнее задание

Контейнеры

Вариант №1

Карточка иностранного слова представляет собой структуру, содержащую иностранное слово и его перевод. Для моделирования электронного словаря иностранных слов реализовать класс Dictionary. Данный класс имеет поле – название словаря и содержит массив структур WordCard, представляющих собой карточки иностранного слова. Название словаря задается при создании нового словаря, но должна быть предоставлена возможность его изменения во время работы. Карточки добавляются в словарь и удаляются из него. Реализовать поиск определенного слова как отдельный метод. Аргументом операции индексирования должно быть иностранное слово. В словаре не должно быть карточек-дублей. Реализовать операции объединения, пересечения и вычитания словарей. При реализации должен создаваться новый словарь, а исходные словари не должны изменяться. При объединении новый словарь должен содержать без повторений все слова, содержащиеся в обоих словарях-операндах. При пересечении новый словарь должен состоять только из тех слов, которые имеются в обоих словарях-операндах. При вычитании новый словарь должен содержать слова первого словаря-операнда, отсутствующие во втором.

Вариант №2

Реализовать класс Realty, моделирующий риэлторскую контору (купля-продажа жилья). Список квартир на продажу состоит из элементов со структурой: район, адрес, количество комнат, общая площадь, жилая площадь, год постройки дома, запрашиваемая стоимость, телефон продавца. Список потребностей состоит из элементов со структурой: требуемое количество комнат, требуемая площадь, предлагаемая цена и телефон покупателя. Реализовать методы поиска вариантов в списках спроса и предложения для удовлетворения спроса. Для обоих списков реализовать обычные операции добавления и удаления. Реализовать операцию генерации конкретного объекта Group (группа), содержащего список предложений, из объекта типа Realty. Должна быть обеспечена возможность выбирать группу по любому из полей класса Realty.

Вариант №3

Реализовать модель станции техобслуживания автомобилей. Один элемент очереди — автомобиль — представляет собой структуру с полями: марка автомобиля, требуемая марка бензина, объем бака, остаток бензина, объем масла, необходимость мойки. Марка автомобиля и марка бензина представляются статическими массивами строк, а в элементе списка эти поля

представлены индексами. Станция техобслуживания предоставляет следующий набор услуг: заправка бензином, заливка масла, мойка. Каждая услуга имеет цену: заливка масла и мойка фиксированную, бензин — цену за литр. Время заливки бензина — 2 мин, заливки масла — 1 мин, мойки — 3 мин. Автомобили прибывают на станцию в случайные моменты времени, в среднем — раз в 5 мин. Требуемые услуги также генерируются случайным образом. Промоделировать работу станции за сутки.

Вариант №4

Создать класс `ListPayment` (зарплата). В классе содержится список сотрудников, для которых рассчитывается заработная плата. Сотрудник представлен классом `Person` с полями: табельный номер, фамилия-имя-отчество, оклад, год поступления на работу, процент надбавки, подоходный налог, количество отработанных дней в месяце, количество рабочих дней в месяце, начисления, удержания. Реализовать методы вычисления класса `Person`: начисленной суммы, удержанной суммы, суммы, выдаваемой на руки, и стажа. Стаж вычисляется как полное количество лет, прошедших от года поступления на работу, до текущего года. Начисления представляют собой сумму, начисленную за отработанные дни и надбавки — доли от суммы, начисленной за отработанные дни. Удержания представляют собой отчисления в пенсионный фонд (1 % от начисленной суммы) и подоходный налог. Подоходный налог составляет 13 % от начисленной суммы без отчислений в пенсионный фонд. Реализовать методы добавления сотрудника в список и удаления из него; методы объединения списков; методы поиска по полям класса `Person`. Реализовать методы вычисления полных сумм по всему списку: начислено, удержано, на руки, подоходный налог, пенсионный фонд.

Вариант №5

Реализовать очередь `Unemployed` на бирже труда. Один элемент списка безработных имеет поля: фамилия, возраст, пол, образование, профессия, должность, стаж работы в последней должности и общий стаж, дата постановки на учет, желаемая заработная плата, желаемая должность. Для реализации поля даты использовать упрощенную версию класса `Date`. Образование, профессия и должность должны быть представлены в классе статическими массивами строк, а в элементе списка безработных — как индексы в этих массивах. База вакансий должна быть представлена классом `Vacancy`, в котором содержится список фирм с полями: название фирмы, вакантная должность, заработная плата, требования к кандидату: образование, возраст, пол, общий стаж и стаж работы в аналогичной должности. Для обоих списков реализовать обычные операции добавления и удаления. Реализовать методы поиска в очереди и в базе вакансий для осуществления снятия с учета на бирже.

Вариант №6

Нагрузка преподавателя за учебный год представляет собой список дисциплин, которые он должен прочитать в течение года. Одна дисциплина представляется информационной структурой с полями: название дисциплины, семестр проведения, количество студентов, количество часов аудиторных лекций, количество аудиторных часов практики, вид контроля (зачет или экзамен). Реализовать класс `WorkTeacher`, моделирующий бланк назначенной преподавателю нагрузки. Класс содержит фамилию преподавателя, дату утверждения, список преподаваемых дисциплин, объем полной нагрузки в часах и в ставках. Дисциплины в списке не должны повторяться. Объем в ставках вычисляется как частное от деления объема в часах на среднюю годовую ставку, одинаковую для всех преподавателей кафедры. Элемент списка преподаваемых дисциплин содержит поля: дисциплина, количество часов, выделяемых на зачет (0,35 ч на одного студента) или экзамен (0,5 ч на студента), сумму часов по дисциплине. Реализовать добавление и удаление дисциплин; вычисление суммарной нагрузки в часах и ставках. Должен осуществляться контроль превышения нагрузки (допустимый максимум — полуторы ставки).

Вариант №7

Товарный чек содержит список товаров, купленных покупателем в магазине. Один элемент списка представляет собой пару: товар-сумма. Товар реализовать как класс `Goods` с полями кода и наименования, цены за единицу товара, количества приобретаемых единиц. В классе должны быть методы доступа к полям с целью получения и изменения информации, а также метод вычисления суммы оплаты товара. Реализовать класс `Receipt`, полями которого являются номер товарного чека, дата и время его создания, список покупаемых товаров. В классе `Receipt` реализовать методы добавления, изменения и удаления записи о покупаемом виде товара, метод поиска информации об определенном виде товара по его коду и названию, а также метод подсчета общей суммы, на которую были осуществлены покупки.

Вариант №8

Информационная запись о файле в каталоге содержит поля: имя файла, расширение, дата и время создания, атрибуты «только чтение», «скрытый», «системный», размер файла на диске. Реализовать класс `Directory`, содержащий название родительского каталога, количество файлов в каталоге, список файлов в каталоге. Один элемент списка содержит информационную запись о файле, дату последнего изменения, признак выделения и признак удаления. Реализовать методы добавления файлов в каталог и удаления файлов из него; метод поиска файла по имени, по расширению, по дате создания; метод вычисления полного объема каталога. Реализовать операцию объединения и операцию пересечения каталогов.

Вариант №9

Информационная запись о книге в библиотеке содержит следующие поля: автор, название, год издания, издательство, цена. Для моделирования учетной карточки абонента реализовать класс `Subscriber`, содержащий фамилию абонента, его библиотечный номер и список взятых в библиотеке книг. Один элемент списка содержит информационную запись о книге, дату выдачи, требуемую дату возврата и признак возврата. Реализовать методы добавления книг в список и удаления книг из него; метод поиска книг, подлежащих возврату; методы поиска по автору, издательству и году издания; метод вычисления стоимости всех подлежащих возврату книг. Реализовать операцию слияния двух учетных карточек, операцию пересечения и вычисления разности. Реализовать операцию генерации конкретного объекта `Debt` (долг), содержащего список книг, подлежащих возврату из объекта типа `Subscriber`.

Вариант №10

Прайс-лист компьютерной фирмы представляет собой список моделей продаваемых компьютеров. Один элемент списка (`Model`) содержит информацию о марке компьютера, типе процессора, частоте работы процессора, объеме памяти, объеме жесткого диска, объеме памяти видеокарты, цене компьютера в условных единицах и количестве экземпляров, имеющихся в наличии. Реализовать класс `PriceList`, полями которого являются дата его создания, номинал условной единицы в рублях и список продаваемых моделей компьютеров. В списке не должно быть двух моделей одинаковой марки. В классе `Pricelist` реализовать методы добавления, изменения и удаления записи о модели, метод поиска информации о модели по марке компьютера, по объему памяти, диска и видеокарты (равно или не меньше заданного), а также метод подсчета общей суммы. Реализовать методы объединения и пересечения прайс-листов. Метод поиска возвращает объект класса `Model` в качестве результата.

Вариант №11

Реализовать очередь `Queue` нуждающихся в улучшении жилищных условий. Элементом очереди является структура `Lodger` с полями: фамилия, количество членов семьи, дата постановки в очередь, занимаемая жилая площадь, необходимые жилищные условия (количество комнат и требуемая площадь). Для представления дат реализовать класс `Date`. Реализовать операцию поиска элемента по фамилии, по дате (после заданной), по требуемой площади (не меньше заданной); реализовать методы постановки в очередь и удаления из очереди. Реализовать операции объединения и пересечения двух очередей (если очередник есть в обеих, включать только один раз).

Вариант №12

Карточка персоны содержит фамилию и дату рождения. Реализовать класс `ListPerson` для работы с картотекой персоналий. Класс должен содержать массив карточек персон. Реализовать методы добавления и удаления карточек, а также метод доступа к карточке по фамилии. Фамилии в массиве должны быть уникальны. Реализовать операции объединения двух картотек, операции пересечения и вычисления разности. Реализовать метод, выдающий по фамилии знак зодиака. Для этого в классе должен быть объявлен статический массив структур `Zodiac` с полями: название знака зодиака, дата начала и дата окончания периода. Индексом в массиве должен быть перечислимый тип `zodiac`.

Вариант №13

Одна запись в списке запланированных дел представляет собой структуру `DailyItem`, которая содержит время начала и окончания работы, описание и признак выполнения. Реализовать класс `DailySchedule` — план работ на день. Реализовать методы добавления, удаления и изменения планируемой работы. При добавлении проверять корректность временных рамок (они не должны пересекаться с уже запланированными мероприятиями). Реализовать метод поиска свободного промежутка времени. Условие поиска задает размер искомого интервала, а также временные рамки, в которые он должен попадать. Метод поиска возвращает структуру `DailyItem` с пустым описанием вида работ. Реализовать операцию генерации объекта `Redo` (еще раз), содержащего список дел, не выполненных в течение дня, из объекта типа `DailySchedule`. Реализовать операцию объединения двух объектов типа `DailySchedule`; в объединенном объекте не должно быть одинаковых работ, выполняемых в разное время, а также разных работ, выполняемых в одно время.

Вариант №14

Список абонентов кабельного телевидения состоит из элементов следующей структуры: фамилия, район, адрес, телефон, номер договора, дата заключения договора, оплата установки, абонентская плата ежемесячно, дата последнего платежа. Реализовать класс `Abonent` в котором предусмотреть методы добавления абонентов в список и удаления их из него; методы поиска абонента по номеру договора, номеру телефона и по фамилии, по дате заключения договора; метод вычисления стоимости платежей одного абонента и всего списка. Реализовать операцию объединения и операцию пересечения списков. Реализовать операцию генерации конкретного объекта `Group` (группа), содержащего список абонентов одного района, из объекта типа `Abonent`.

Вариант №15

Реализовать класс `Bill`, представляющий собой разовый платеж за разговор. Класс содержит следующие поля: фамилия плательщика, номер телефона,

тариф за минуту разговора, скидка (в процентах), время начала разговора, время окончания разговора, сумма к оплате. Реализовать методы извлечения и изменения полей. Время разговора, подлежащее к оплате, вычисляется в минутах; неполная минута считается за полную; реализовать метод, возвращающий сумму в рублях. Используя класс `Bill`, реализовать класс `ListPayer`. Класс содержит список плательщиков за телефонные услуги, дату создания списка, номер списка. Один элемент списка включает информацию о плательщике (класс `Bill`), статус оплаты, дату платежа, сумму платежа. Реализовать методы добавления плательщиков в список и удаления их из него; метод поиска плательщика по номеру телефона и по фамилии, по дате платежа. Метод вычисления полной стоимости платежей всего списка. Реализовать операцию объединения и операцию пересечения списков.

Вариант №16

Один тестовый вопрос представляет собой структуру `Task` со следующими полями: вопрос, пять вариантов ответа, номер правильного ответа, начисляемые баллы за правильный ответ. Для моделирования набора тестовых вопросов реализовать класс `TestContent`, содержащий массив тестовых вопросов. Реализовать методы добавления и удаления тестовых вопросов, а также метод доступа к тестовому заданию по его порядковому номеру в списке. В массиве не должно быть повторяющихся вопросов. Реализовать операцию слияния двух тестовых наборов, операции пересечения и вычисления разности. Реализовать операцию генерации конкретного объекта `Test` объемом не более `k` вопросов из объекта типа `TestContent`.

Вариант №17

Учебный план специальности является списком дисциплин, которые студент должен изучить за время обучения. Одна дисциплина представляет собой структуру с полями: номер дисциплины в плане, тип дисциплины (федеральная, региональная, по выбору), название дисциплины, семестр, в котором дисциплина изучается, вид итогового контроля (зачет или экзамен), общее количество часов, необходимое для изучения дисциплины, количество аудиторных часов, которые состоят из лекционных часов и часов практики. Реализовать класс `PlanEducation` для моделирования учебного плана специальности. Класс должен содержать код и название специальности, дату утверждения, общее количество часов специальности по стандарту и список дисциплин. Один элемент списка дисциплин должен содержать запись о дисциплине, количество часов самостоятельной работы (разность между общим количеством часов и аудиторными часами), признак наличия курсовой работы, выполняемой по данной дисциплине. Реализовать методы добавления и удаления дисциплин; метод поиска дисциплины по семестру, по типу

дисциплины, по виду итогового контроля; метод вычисления суммарного количества часов всех дисциплин; метод вычисления количества экзаменов и зачетов по семестрам.

Вариант №18

Реализовать класс Library (библиотека модулей). Библиотека включает в себя поле количества модулей. Реализовать методы добавления, удаления и замены модуля/группы модулей в библиотеке; метод объединения, пересечения библиотек; метод вычисления общего объема модулей в библиотеке; метод поиска модуля по имени. Каждый модуль имеет имя (не более 30 символов) и объем в байтах. Модули в библиотеке не должны повторяться.