В этой теме в каждом упражнении требуется реализовать в том или ином виде определение нового класса. Во всех заданиях необходимо реализовать конструк­тор инициализации (один или несколько) и конструктор без аргументов. Ука­занные в задании операции реализуются посредством перегрузки подходящих операций. Во всех заданиях обязательно должны быть реализованы соответ­ствующие операции с присваиванием, ввод с клавиатуры, вывод на экран, преоб­разования в строку toString.

Дополнительно к требуемым в заданиях операциям перегрузить операцию индексирования []. Максимально возможный размер массива задать константой. В отдельном поле size должно храниться максимальное для данного объекта ко­личество элементов массива; реализовать метод sizeO, возвращающий установ­ленную длину. Если количество элементов массива изменяется во время работы, определить в классе поле count. Первоначальные значения size и count устанавливаются конструктором.

1. Создать класс BitString для работы с битовыми строками не более чем из 100 бит. Битовая строка должна быть представлена массивом типа unsigned char, каждый элемент которого принимает значение 0 или 1. Реальный размер массива задается как аргумент конструктора инициализации. Должны быть реализованы все традиционные операции для работы с битовыми строками: and, or, xor, not. Реализовать сдвиг влево и сдвиг вправо на заданное количест­во битов.
2. Создать класс Decimal для работы с беззнаковыми целыми десятичными чис­лами, используя для представления числа массив из 100 элементов типа unsigned char, каждый из которых является десятичной цифрой. Младшая цифра имеет меньший индекс (единицы — в нулевом элементе массива). Ре­альный размер массива задается как аргумент конструктора инициализации. Реализовать арифметические операции, аналогичные встроенным для целых в C++, и операции сравнения.
3. Создать класс Hex для работы с беззнаковыми целыми шестнадцатеричными числами, используя для представления числа массив из 100 элементов типа unsigned char, каждый из которых является шестнадцатеричной цифрой. Младшая цифра имеет меньший индекс. Реальный размер массива задается как аргумент конструктора инициализации. Реализовать арифметические операции, аналогичные встроенным для целых в C++, и операции сравнения.
4. Создать класс Money для работы с денежными суммами. Сумма должна быть представлена массивом, каждый элемент которого — десятичная цифра. Максимальная длина массива — 100 цифр, реальная длина задается конструктором. Младший индекс соответствует младшей цифре денежной суммы. Младшие две цифры — копейки.
5. Создать класс Polinom для работы с многочленами до 100-й степени. Коэффициенты должны быть представлены массивом из 100 элементов-коэффициентов. Младшая степень имеет меньший индекс (нулевая степень — нулевой индекс). Размер массива задается как аргумент конструктора инициализации. Реализовать арифметические операции и операции сравнения, вычисление значения полинома для заданного значения х, дифференцирование, интегрирование.
6. Создать класс Fraction для работы с беззнаковыми дробными десятичными числами. Число должно быть представлено двумя массивами типа unsigned char: целая и дробная часть, каждый элемент — десятичная цифра. Для целой части младшая цифра имеет меньший индекс, для дробной части старшая цифра имеет меньший индекс (десятые — в нулевом элементе, сотые — в первом, и т. д.). Реальный размер массивов задается как аргумент конструктора инициализации. Реализовать арифметические операции сложения, вычита­ния и умножения, и операции сравнения.
7. Реализовать класс Rational, используя два массива из 100 элементов типа unsigned char для представления числителя и знаменателя. Каждый элемент является десятичной цифрой. Младшая цифра имеет меньший индекс (единицы — в нулевом элементе массива). Реальный размер массива задается как аргумент конструктора инициализации.
8. 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.
9. 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, операцию пересечения и вычисления разности (см. задание 8). Дополнительно реализовать операцию генерации конкретного объекта Test объемом не более К вопросов из объекта типа TestContent.
10. 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 персон, а также метод доступа к карточке по фамилии. Фамилии в массиве должны быть уникальны. Реализовать операции объединения двух картотек, операцию пересечения и вычисления разности (см. задание 8). Реализовать метод, выдающий по фамилии знак зодиака. Для этого в классе должен быть объявлен массив структур Zodiac с полями: название знака зодиака, дата начала и дата окончания периода. Индексом в массиве должен быть перечислимый тип zodiас. Реализовать два варианта класса: с обычным массивом и статическим массивом Zodiac.
11. Товарный чек содержит список товаров, купленных покупателем в магазине. Один элемент списка представляет собой пару: товар-сумма. Товар — это класс Goods с полями кода и наименования товара, цены за единицу товара, количества покупаемых единиц товара. В классе должны быть реализованы методы доступа к полям для получения и изменения информации, а также метод вычисления суммы оплаты за товар. Для моделирования товарного чека 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 товаре, метод поиска информации об определенном виде товара по его коду, а также метод подсчета общей суммы, на которую были осуществлены покупки. Методы добавления и изменения принимают в качестве аргумента объект класса Goods. Метод поиска возвращает объект класса Goods в качестве результата.
12. 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 состоит из информационной записи о книге, даты выдачи, требуемой даты возврата и признака 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 (см. задание 8). Реализовать операцию генерации конкретного объекта Debt (долг), содержащего список книг, подлежащих возврату из объекта типа Subscriber.
13. Информационная запись о файле в каталоге содержит поля: имя файла, расширение, дата и время создания, атрибуты «только чтение», «скрытый», «системный», размер файла на диске. Для моделирования каталога реализовать класс Directory, содержащий название родительского каталога, количество файлов в каталоге, список файлов в каталоге. Один элемент списка включает в себя 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 (см. задание 8). Реализовать операцию генерации конкретного объекта Group (группа), содер­жащего список файлов, из объекта типа Directory. Должна быть возможность выбирать группу файлов по признаку удаления, по атрибутам, по дате создания (до. или после), по объему (меньше или больше).
14. 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 для 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. Реализовать операцию объединения, операцию вычитания учебных планов и операцию пересечения учебных планов (см. задание 8). Реализовать операцию генерации конкретного объекта Group (группа дисциплин), содержащего список дисциплин, из объекта типа PlanEducation. Должна быть возможность выбирать группу дисциплин по типу, по семестру, по виду итогового контроля, по наличию курсовой работы. Должен осуществляться контроль за суммарным количеством часов, за количеством экзаменов в семестре (не более пяти и не менее трех).
15. Нагрузка преподавателя за учебный год представляет собой список дисциплин, преподаваемых им 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 дисциплину, 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 за превышени­ем нагрузки более полутора ставок.
16. Создать класс Octal для работы с беззнаковыми целыми восьмеричными числами, используя для представления числа массив из 100 элементов типа unsigned char, каждый элемент которого является восьмеричной цифрой. Младшая цифра имеет меньший индекс (единицы — в нулевом элементе массива). Реальный размер массива задается как аргумент конструктора инициализации. Реализовать арифметические операции, аналогичные встроенным для целых в C++, и операции сравнения.
17. Создать класс String для работы со строками, аналогичными строкам Turbo Pascal (строка представляется как массив 255 байт, длина — в первом байте). Максимальный размер строки должен задаваться. Обязательно должны быть реализованы: определение длины строки, поиск подстроки в строке, удаление подстроки из строки, вставка подстроки в строку, сцепление двух строк.
18. Одна запись в списке запланированных дел представляет собой структуру Dailytern, которая содержит время начала и окончания работы, описание и признак выполнения. Реализовать класс DailySchedule, представляющий

собой 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 Dailytern с пустым описанием вида работ. Реализовать операцию генерации объекта Redo (еще раз), содержащего список дел, не выполненных в течение дня, из объекта типа DailySchedule

1. Прайс-лист компьютерной фирмы включает в себя список моделей продаваемых компьютеров. Одна позиция списка (Model) содержит марку компьютера, тип процессора, частоту работы процессора, объем памяти, объем жесткого диска, объем памяти видеокарты, цену компьютера в условных единицах и количество 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 (см. задание 8). Методы добавления и изменения принимают в качестве входного параметра объект класса Model. Метод поиска возвращает объект класса Model в качестве результата.
2. Реализовать класс Set (множество) не более чем из 256 элементов-символов. Решение должно обеспечивать включение элемента в множество, исключение элемента из множества, объединение, пересечение и разность множеств, вычисление количества элементов в множестве, проверку присутствия элемента в множестве, проверку включения одного множества в другое.