Структуры и классы

Во всех заданиях, помимо указанных в задании операций, обязательно должны быть реализованы следующие методы:

* метод инициализации Init;
* ввод с клавиатуры Read;
* вывод на экран Display;
* преобразование в строку toString.

Все задания должны быть реализованы двумя способами:

* тип данных представляется структурой с необходимыми полями, а операции реализуются как внешние функции, которые получают объекты данного типа в качестве аргументов;
* как класс с закрытыми полями, где операции реализуются как методы класса;

1. Комплексное число представляются парой действительных чисел (а, b), где а — действительная часть, b — мнимая часть. Реализовать класс Complex для работы с комплексными числами. Обязательно должны присутствовать опе­рации:

сложения add, (а, Ь) + (с, d) - (а + Ь, с + d);

вычитания sub, (а, Ь) - (с, d) = (а - Ь, с - d);

умножения mul, (а, Ь) х (с, d) = (ас - bd, ad + be);

деления di v, (a, b) / (с, d) = (ас + bd, be - ad) / (с2 + d2);

сравнение equ, (a, b) = (c, d), если (a = с) и (b = d);

сопряженное число conj, conj(a, b) = (a, -b).

1. Создать класс vector3D, задаваемый тройкой координат. Обязательно должны быть реализованы: сложение и вычитание векторов, скалярное произведение векторов, умножение на скаляр, сравнение векторов, вычисление длины век­тора, сравнение длины векторов.
2. Создать класс Model Window для работы с моделями экранных окон. В качестве полей задаются: заголовок окна, координаты левого верхнего угла, размер по горизонтали, размер по вертикали, цвет окна, состояние «видимое/невиди­мое», состояние «с рамкой/без рамки». Координаты и размеры указываются в целых числах. Реализовать операции: передвижение окна по горизонтали, по вертикали; изменение высоты и/или ширины окна изменение цвета; изме­нение состояния, опрос состояния. Операции передвижения и изменения раз­мера должны осуществлять проверку на пересечение границ экрана. Функция вывода на экран должна индуцировать состояние полей объекта.
3. Создать класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа unsigned char — для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение, вычитание, деление сумм, де­ление суммы на дробное число, умножение на дробное число и операции сравнения.
4. Создать класс Triangle для представления треугольника. Поля данных долж­ны включать углы и стороны. Требуется реализовать операции: получения и изменения полей данных, вычисления площади, вычисления периметра, вычисления высот, а также определения вида треугольника (равносторонний, равнобедренный или прямоугольный).
5. Создать класс Angle для работы с углами на плоскости, задаваемыми величи­ной в градусах и минутах. Обязательно должны быть реализованы: перевод в радианы, приведение к диапазону 0-360, увеличение и уменьшение угла на заданную величину, получение синуса, сравнение углов.
6. Создать класс Point для работы с точками на плоскости. Координаты точки — декартовы. Обязательно должны быть реализованы: перемещение точки по оси X, перемещение по оси У, определение расстояния до начала координат, расстояния между двумя точками, преобразование в полярные координаты, сравнение на совпадение и несовпадение.
7. Рациональная (несократимая) дробь представляется парой целых чисел (а, b), где а — числитель, b — знаменатель. Создать класс Rational для работы с ра­циональными дробями. Обязательно должны быть реализованы операции:

сложения add, (а, Ь) + (с, et) = (ad + be, bet);

вычитания sub, (a, b) - (с, d) = (ad - be, bet);

умножения mul, (a, b) x (c, et) - (ac, bet);

деления div, (a, b) / (c, et) = (ad, be);

сравнения equal, greate, less.

Должна быть реализована приватная функция сокращения дроби reduce, ко­торая обязательно вызывается при выполнении арифметических операций.

1. Создать класс Date для работы с датами в формате «год.месяц.день». Дата представляется структурой с тремя полями типа unsigned int: для года, месяца и дня. Класс должен включать не менее трех функций инициализации: числа­ми, строкой вида «год.месяц.день» (например, «2004.08.31») и датой. Обя­зательными операциями являются: вычисление даты через заданное коли­чество дней, вычитание заданного количества дней из даты, определение високосности года, присвоение и получение отдельных частей (год, месяц, день), сравнение дат (равно, до, после), вычисление количества дней между датами.
2. Создать класс Time для работы со временем в формате «час:минута:секунда». Класс должен включать в себя не менее четырех функций инициализации: числами, строкой (например, «23:59:59»), секундами и временем. Обязатель­ными операциями являются: вычисление разницы между двумя моментами времени в секундах, сложение времени и заданного количества секунд, вычи­тание из времени заданного количества секунд, сравнение моментов времени, перевод в секунды, перевод в минуты (с округлением до целой минуты).
3. Реализовать класс FazzyNumber для работы с нечеткими числами, которые представляются тройками чисел (х - в\, х, х + ег). Для чисел А = (А - а/, А, А + аг) и В = (В - bi, В, В + Ьг) арифметические операции выполняются по следующим формулам:

О сложение А + В = (Л + В - щ - bi, А + В, А + В + ar + Ьг);

О вычитание А - В = (А - В - а/ - 6/, А - В, А - В + ar + Ьг);

О умножение АВ = (A xB-Bxai- Ах bi + aixbi, Ах В, АхВ + Вхсц + + А х bi + at х bi);

О обратное число А = (1 / (А + ar), 1 / А, 1 / (А - а;)), А > 0;

О деление А / В = ((Л - at) / (В + Ьг), А / В, (А + аг) / (В - bi)), В > 0;

1. Реализовать класс Account, представляющий собой банковский счет. В классе должны быть четыре поля: фамилия владельца, номер счета, процент начис­ления и сумма в рублях. Открытие нового счета выполняется операцией ини­циализации. Необходимо выполнять следующие операции: сменить владель­ца счета, снять некоторую сумму денег со счета, положить деньги на счет, начислить проценты, перевести сумму в доллары, перевести сумму в евро, по­лучить сумму прописью (преобразовать в числительное).
2. Номиналы российских рублей могут принимать значения 1, 2, 5, 10, 50, 100, 500, 1000, 5000. Копейки представить как 0.01 (1 копейка), 0.05 (5 копеек), 0.1 (10 копеек), 0.5 (50 копеек). Создать класс Money для работы с денежны­ми суммами. Сумма должна быть представлена полями-номиналами, значе­ниями которых должно быть количество купюр данного достоинства. Реали­зовать сложение сумм, вычитание сумм, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения. Дробная часть (копейки) при выводе на экран должны быть отделена от целой части запятой.
3. Реализовать класс Bankomat, моделирующий работу банкомата. В классе долж­ны содержаться поля для хранения идентификационного номера банкомата, информации о текущей сумме денег, оставшейся в банкомате, минимальной и максимальной суммах, которые позволяется снять клиенту в один день. Сум­ма денег представляется полями-номиналами 10-1000 (см. задание 13). Реа­лизовать метод инициализации банкомата, метод загрузки купюр в банкомат и метод снятия определенной суммы денег. Метод снятия денег должен вы­полнять проверку на корректность снимаемой суммы: она не должна быть меньше минимального значения и не должна превышать максимальное зна­чение. Метод toString должен преобразовать в строку сумму денег, остав­шуюся в банкомате.
4. Создать класс Fraction для работы с дробными числами. Число должно быть представлено двумя полями: целая часть — длинное целое со знаком, дробная часть — беззнаковое короткое целое. Реализовать арифметические операции сложения, вычитания, умножения и операции сравнения.
5. Создать класс Goods (товар). В классе должны быть представлены поля: на­именование товара, дата оформления, цена товара, количество единиц товара, номер накладной, по которой товар поступил на склад. Реализовать методы изменения цены товара, изменения количества товара (увеличения и умень­шения), вычисления стоимости товара. Метод toStringO должен выдавать в виде строки стоимость товара.
6. Создать класс BitString для работы с 64-битовыми строками. Битовая строка должна быть представлена двумя полями типа unsigned long. Должны быть реализованы все традиционные операции для работы с битами: and, or, xor, not. Реализовать сдвиг влево shiftLeft и сдвиг вправо shiftRight на заданное количество битов.
7. Создать класс LongLong для работы с целыми числами из 64 бит. Число долж­но быть представлено двумя полями: long — старшая часть, unsigned long — младшая часть. Должны быть реализованы арифметические операции, при­сутствующие в C++ (без присваивания), и сравнения.
8. Создать класс Payment (зарплата). В классе должны быть представлены поля: фамилия-имя-отчество, оклад, год поступления на работу, процент надбав­ки, подоходный налог, количество отработанных дней в месяце, количество рабочих дней в месяце, начисленная и удержанная суммы. Реализовать мето­ды: вычисления начисленной суммы, вычисления удержанной суммы, вычис­ления суммы, выдаваемой на руки, вычисления 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, то есть доли от первой суммы. 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.
9. Реализовать класс Cursor. Полями являются координаты курсора по горизон­тали и вертикали — целые положительные числа, вид курсора — горизонталь­ный или вертикальный, размер курсора — целое от 1 до 15. Реализовать мето­ды изменения координат курсора, изменения вида курсора, изменения размера курсора, метод гашения и восстановления курсора.