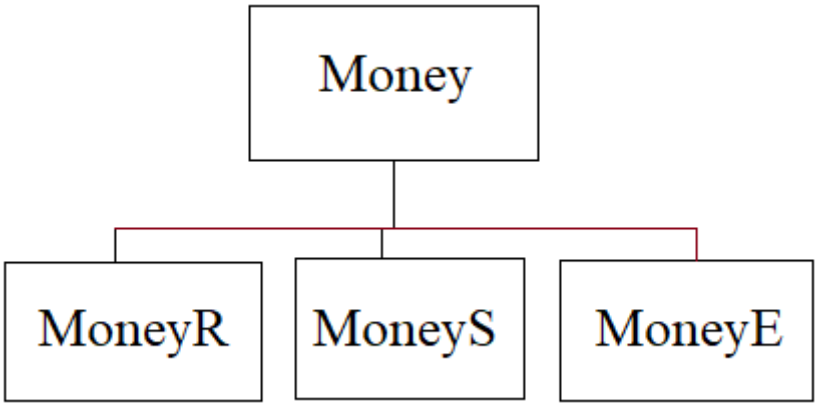
Задания по ООП C++ для предмета «Языки программирования» 1-й семестр

1. Создать шаблонный класс (для использования разных типов данных) Complex работы с комплексными числами. Класс должен уметь складывать, вычитать, делить и умножать комплексные числа (используя операции перегрузки соответствующих операторов). Также в классе необходимо реализовать методы для вычисления модуля числа и его аргумента (угла). При невозможности выполнения тех или иных действий (например, деление на нуль), класс должен генерировать соответствующие исключения.

2. Создать класс MyString для работы со строками. Класс должен хранить указатель типа char * на динамический массив символов, в котором хранится текущая строка. Реализовать перегрузку операторов: присваивания, сложения (соединения) строк, сравнения (== и !=). Реализовать проверку на корректность вносимых данных и генерацию исключений при некорректной работе с классом.

3. Разработать шаблонный класс List связного списка (шаблон будет указывать тип хранимых данных). Реализовать методы: push_back (добавление в конец списка), push_front (добавление в начало списка), pop_back (удаление последнего элемента), pop_front (удаление первого элемента), isEmpty (проверка списка на пустоту), size (возвращает число элементов), show (отображение списка на экране), getAt(n) (получение n-го элемента списка).

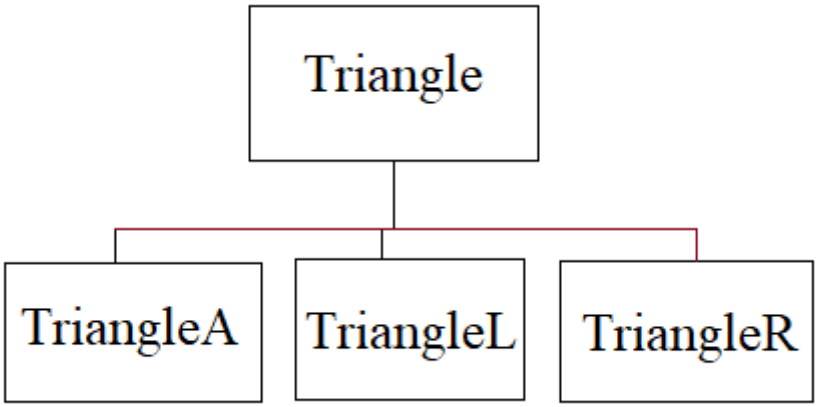
4. Класс для работы с различными денежными суммами. Программа должна иметь следующую иерархию объектов:



В базовом классе Money должны быть описаны общие данные, методы и/или операторы работы с денежными единицами, а в дочерних классах – перегруженные методы и/или операторы для работы с конкретной денежной единицей.

Денежные единицы должны быть представлены двумя полями: типа int (рубли, доллары, евро) и типа char (копейки, центы, центы). Дробная часть (копейки, центы, центы) при выводе на экран должна быть отделена от целой части запятой. Реализовать с использованием перегрузки операторов: сложение, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения.

5. Реализовать шаблонный класс Triangle для представления треугольников. Шаблон определяет тип координат треугольника. Программа должна иметь следующую иерархию объектов:



В базовом классе Triangle реализовать общие данные, методы и/или операторы работы с треугольниками. Шаблонные классы: TriangleA предназначен для представления равносторонних треугольников, TriangleL – прямоугольных, TriangleR – равнобедренных.

Реализовать методы для вычисления периметра, площади, вывод на экран информации о длине каждой из сторон. В качестве данных в классе хранятся координаты вершин треугольника.

6. Реализовать шаблонный класс DArray для работы с динамическим массивом. Шаблон должен определять тип элементов этого массива. В классе должны быть реализованы операторы: присваивания одного массива другому, сложение (добавление) другого массива, сравнения массивов (== и !=). Должны быть реализованы методы: получения числа элементов массива, изменения размера массива, вставки и удаления произвольного элемента в массиве, изменения значения существующего элемента. Также методы и операторы класса должны генерировать исключения при некорректной работе (например, записи значения в несуществующий элемент массива).

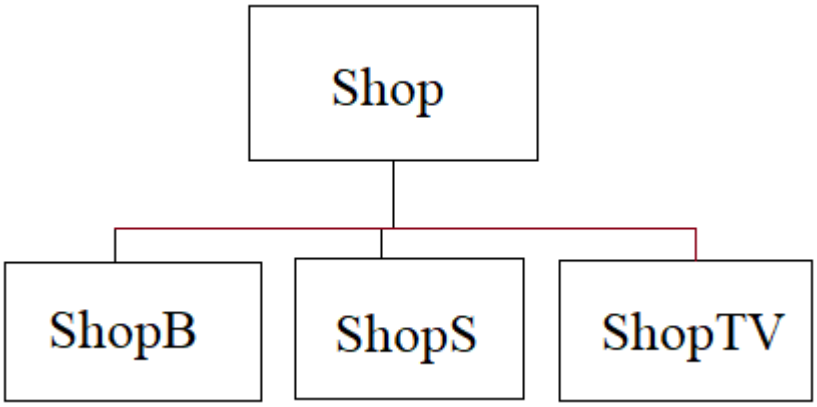
7. Создать класс CFile для работы с файлами, используя функции языка C: fopen, fclose, fputs, fgets, fprintf, fscanf, fwrite, fread, ftell, fseek. Класс должен реализовывать методы: открыть (open) и закрыть (close) файл (как в текстовом режиме, так и в бинарном), записать n байт, прочитать n байт. Реализовать операторы << и >> для записи и считывания строки из файла. Критические методы и операторы должны генерировать исключения (throw) при соответствующих проверках.

8. Написать класс Clock представления времени (часы, минуты, секунды). Класс должен реализовывать операторы: сложение (для сложения времени), вычитание (для вычитания времени), сравнения (== и !=), присваивания (для задания времени). Также должен быть реализован метод отображения текущего времени в разных форматах: hh:mm:ss, hh.mm.ss, hh/mm/ss, h:m:s, h.m.s, h/m/s (Здесь h – часы, m – минуты, s – секунды. Формат hh:mm:ss означает, что должен быть значащий 0 у

цифр, например, 01:01:01. Формат h:m:s означает, что значащих нулей нет, например, 2:1:3.). Если пользователь вводит неверный формат для времени через методы, операторы или в конструкторе, то класс Clock должен генерировать исключение.

9. Написать шаблонный класс Matrix для работы с матрицами. Шаблон определяет тип элементов матрицы. Элементы матрицы должны храниться в виде динамического массива. Сам класс должен реализовывать операторы: присваивания, сложения, вычитания, умножения матриц. В случаях невозможности вычисления матриц класс Matrix должен генерировать соответствующие исключения. В классе должны быть реализованы методы: получение размеров матрицы, получение элемента массива с индексами i,j, запись значения в элемент с индексами i,j, транспонирование матрицы. Обратить внимание, при описании класса задать конструктор копирования, а в деструкторе делать освобождение памяти динамического массива.

10. Реализовать класс магазина Shop в виде следующей иерархии:



В базовом классе Shop хранить все общие данные, методы, операторы работы с магазином. Дочерние классы ShopB для торговли книгами, ShopS – смартфонами, ShopTV – телевизорами. Данный магазин должен содержать методы: добавление/удаление товара из корзины, оплата товаров в корзине, получение числа товаров в корзине, вывод на экран списка товаров корзины, получение информации об оплате (за все время работы магазина (и отдельно по книгам, смартфонам и телевизорам), за последнюю транзакцию – общую сумму и отдельно по каждой категории товаров).

Обратите внимание, что все манипуляции с магазином должны выполняться через методы базового класса Shop. А информация по каждой отдельной категории товаров (например, цена, количество) должна храниться в соответствующих дочерних классах.

11. Создать шаблонный класс трапеции Trap, в котором шаблон определяет тип координат вершин трапеции. Данный класс должен иметь методы: вычисления периметра, площади, длин каждой стороны, вывод на экран координат трапеции. И операторы: присваивания, сравнения (== и !=). Также в этом классе должна быть

статическая переменная для подсчета числа созданных объектов этого класса. Должен быть реализован метод (геттер), возвращающий значение этой статической переменной.

12. Написать класс работы с географическими координатами: широта, долгота, которые представляются в виде: градус, минута, секунда. Данный класс должен реализовывать операторы: присваивания, сложения (координат), вычитание (координат), сравнения (== и !=). Также должны быть методы: вывод координат на экран, получение текущей информации по долготе и широте. Все операторы и методы должны генерировать соответствующие исключения, если формат передаваемых данных является некорректным.

13. Составить описание шаблонного класса Rect для прямоугольников со сторонами, параллельными осям координат. Шаблон определяет тип координат прямоугольника. В классе предусмотреть следующие методы: перемещения на плоскости (translate), изменения масштаба (scale), проверки попадания точки в прямоугольник, получение координат прямоугольника, получение периметра и площади, draw для отображения текущих координат на экран. Реализовать следующие операторы: присваивания, сложения (координаты прямоугольников складываются так, чтобы они образовывали один наибольший прямоугольник), *= для изменения масштаба, проверки (== и !=). Если при инициализации или других операциях координаты не соответствуют прямоугольнику, генерировать исключение.

14. Создать шаблонный класс Vector для работы с векторами в двумерном пространстве. Шаблон указывает тип координат вектора (координат начала и конца). Реализовать методы: скалярного произведения векторов (dot), вычисления нормы вектора (norm), длины вектора (length), вывод координат вектора на экран (draw). Реализовать операторы: присваивания, сложения, умножения (векторного), проверок (== и !=). Если какие-либо операции не могут быть выполнены корректно, то предусмотреть генерацию соответствующих исключений.

15. Реализовать агрегацию классов Pole (поле) и Cell (ячейка) для игры в крестики-нолики, размером n x n ячеек. Для программиста должны быть доступны методы: создание игрового поля (create), размещение в ячейке i, j нолика или крестика, чтение содержимого i, j ячейки (что там находится: нолик или крестик), вывод поля на экран (show), проверки выигрышной комбинации (isWin). Если происходит доступ к несуществующей ячейки, соответствующий метод должен генерировать исключение.