**Лабораторная работа №12. Часть 1**

*«Классы. Наследование»*

**Рекомендации по программированию**

* Поля обязательно делать private.
* Интерфейс должен быть полным, т.е. предоставлять возможность выполнять любые разумные действия с классом. И одновременно минимально необходимым, т.е. без дублирования и пересечения возможностей методов.
* Каждый метод класса должен решать только одну задачу. Не стоит объединять два коротких независимых фрагмента кода в один метод.
* Если действия встречаются хотя бы дважды, стоит оформить их в отдельные функции.
* Все заданные методы, поля и экземпляры классов должны иметь осмысленные имена.

**Требования к лабораторной**

* Каждый разрабатываемый класс должен, как правило, содержать следующие элементы: поля с различными спецификаторами, конструкторы с параметрами и без параметров, методы, свойства.
* Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса.
* К каждому отчету, помимо формулировки задания и скрина консоли, нужно прикрепить еще диаграмму классов

**Задания для самостоятельного решения**

При выполнении данной работы необходимо определить базовый класс и производные от него классы. Предусмотреть передачу аргументов конструкторам базового класса; использование виртуальных и перегруженных функций; обработку исключительных ситуаций.

**Варианты с 1-10**

В следующих заданиях требуется создать базовый класс (как вариант абстрактный базовый класс) и определить свойства get ( ), set ( ), метод show ( ) и другие, специфические для данного класса. Создать производные классы, в которые добавить методы согласно индивидуальному заданию.

Часть методов переопределить. Создать массив объектов базового класса и заполнить объектами производных классов. Объекты произ­водных классов идентифицировать конструктором.

Вызвать метод show ( ) базового класса и просмотреть массив объектов.

1. Создать базовый класс «Транспортное средство» и производные классы «Автомобиль», «Велосипед», «Повозка». Подсчитать время и стоимость пере­возки пассажиров и грузов каждым транспортным средством.

2. Создать базовый класс «Грузоперевозчик» и производные классы «Самолет», «Поезд», «Автомобиль». Определить время и стоимость перевозки для указанных городов и расстояний.

3. Создать аналогичный базовый класс «Пассажироперевозчик» и производные классы «Самолет», «Поезд», «Автомобиль». Определить время и стоимость передвижения.

4. Создать базовый класс «Учащийся» и производные классы «Школьник» и «Студент». Создать массив объектов базового класса и заполнить этот массив объектами. Показать отдельно студентов и школьников.

5. Создать базовый класс «Музыкальный инструмент» и производные классы «Ударный», «Струнный», «Духовой». Создать массив объектов «Оркестр». Выдать состав оркестра, переопределив метод.

6. Создать абстрактный класс «Работник фирмы» и производные классы «Менеджер», «Администратор», «Программист». Создать массив объектов базового класса и заполнить этот массив объектами. Показать отдельно элементы каждого класса. Определить среднюю зп всех сотрудников.

7. Определить базовый класс «Множество» и производный класс «Кольцо» (операции сложения и умножения обе коммутативные и ассоциативные, связанные законом дистрибутивности; сложение обладает обратной операцией – вычитанием). Ввести кольца целых чисел, многочленов, систему классов целых чисел, сравнимых по модулю. Кольцо является полем, если в нем определена операция деления, кроме деления на нуль. Рациональные числа, дробно рациональные функции.

8. Создать абстрактный класс «Магазин» с производным классом «Стеллаж», от которого образовывается еще один класс «Полка». Создать массив объектов базового класса и заполнить этот массив объектами. Показать отдельно элементы каждого класса. Определить количество товаров на каждом стелаже.

9. Создать базовый класс «Домашнее животное» и производные классы «Собака», «Кошка», «Попугай» и др. С помощью конструктора установить имя каждого животного и его характеристики.

10. Создать базовый класс «Садовое дерево» и производные классы «Яблоня», «Вишня», «Груша» и др. Создать массив объектов каждого класса. Принять решение о пересадке каждого дерева в зависимости от возраста и плодоношения.

**Варианты 11-27**

1. Создать класс Item (единица хранения в библиотеке), содержащий данные-члены: invNumber – инвентарный номер и taken – взято на руки или имеется в наличии, а также методы:

virtual void Show(); //показать информацию о единице хранения

bool isAvailable(); // есть ли единица хранения в наличии ?

int GetinvNumber(); //возвращает инвентарный номер

void Take(); // операция «взять»

void Return(); // операция «вернуть»

Построить производные классы Book и Magazin. Класс Book содержит данные-члены: author, title, publisher, year и методы: Author();

Title(); Publisher(); YearOf Publishing(); Show().

Класс Magazin включает данные-члены: volume; number; year; title и методы: Volume(); Title(); Number(); Year(); Show().

2. Создать базовый класс Polygon (многоугольник). Класс должен содержать методы для рисования многоугольника, вычисления периметра, нахождения площади и др. Построить производный класс Triangle (треугольник), содержащий также методы для нахождения точки пересечения медиан, длин медиан, длин биссектрис, координат точек пересечения биссектрис, высот треугольника.

3. Создать абстрактный класс Shape для рисования плоских фигур. Построить производные классы Square (квадрат, который характеризуется координатами левого верхнего угла и длиной стороны), Circle (окружность с заданными координатами центра и радиусом), Ellipse (эллипс с заданными координатами вершин описанного вокруг него прямоугольника), позволяющие рисовать указанные фигуры, а также передвигать их на плоскости.

4. Создать класс CPoint – точка и производные от него классы CcoloredPoint и CLine. На основе классов CcoloredPoint и CLine создать класс CcoloredLine. Все классы должны иметь методы для установки и получения значений всех координат, а также изменения цвета и получения текущего цвета.

5. Описать базовый класс Stroka. Обязательные данные-члены класса: указатель типа char – для хранения строки; значение типа int – длина строки.

Методы: конструктор без параметров; конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом); конструктор копирования; получение длины строки; очистка строки (сделать строку пустой); деструктор.

Описать производный класс «БИТОВАЯ\_СТРОКА» (строки данного класса могут содержать только символы ‘0’ и ‘1’). Если в основе инициализирующей строки встретятся любые символы, отличные от допустимых, то БИТОВАЯ\_СТРОКА становится пустой. Содержимое строки рассматривается как двоичное представление целого числа со знаковым разрядом. Отрицательные числа хранятся в дополнительном коде.

Обязательные методы: конструктор без параметров; конструктор, принимающий в качестве параметра C-строку; конструктор копирования; деструктор; изменение знака числа (перевод числа в дополнительный код).

Переопределить следующие операции (длина строки результата в случае необходимости расширяется влево знаковым разрядом): присваивание; сложение (+); проверка на равенство (==).

6. Создать производный класс «СТРОКА10» (целое неотрицательное десятичное число) от класса «СТРОКА» (описание приведено выше).

Методы: конструктор без параметров; конструктор, принимающий в качестве параметра C-строку; конструктор копирования; деструктор; метод, определяющий, можно ли представить данное число в формате int; метод, определяющий, равно ли число нулю; метод, возвращающий представление числа в виде целого (int); метод, удаляющий незначащие нули.

Переопределить операции: сложение (+); проверка на больше (по значению) (>); проверка на меньше (<); присваивание (=).

7. Создать производный класс «БУЛЕВ ВЕКТОР» (**BoolVector**) от класса **Vector**. Компоненты принимают значения из множе­ства {0,1}.

Методы: конструктор без параметров; конструктор, принимающий в качестве параметров указатель на массив це­лого типа (если эле­менты массива содержат числа, отличные от 0 и 1, то создается пустой вектор) и размер вектора; конструктор копирования; деструктор; метод, возвращающий число единиц в векторе; метод, возвращающий позицию самой левой единицы в векторе.

Переопределить операции: поразрядная конъюнкция (&); поразрядная дизъюнкция (|); поразрядная инверсия (~); поразрядная операция ИСКЛЮЧАЮЩЕЕ ИЛИ (^); присваивание (=).

8. Создать производный класс «ТРОИЧНЫЙ ВЕКТОР» от класса **Vector**. Компоненты принимают значения из множества {0,1,2}.

Методы: конструктор без параметров; конструктор, принимающий в качестве параметров указатель на массив целого типа и размер вектора; конструктор копирования; деструктор; проверка двух векторов на ортогональность (два троичных вектора называются ортогональными, если в них существует пара одноименных компонент, имеющих в одном из векторов значение 0, а в другом – 1); метод, возвращающий число компонент в векторе, принимающих значение 2.

Переопределить операции: присваивание (=); поразрядная конъюнкция (пересечение) двух не ортогональных векторов (&): 0&0=0, 1&1=1, 2&2=2, 0&2=0, 2&0=0, 1&2=1, 2&1=1; индексиро­вание ([]).

9. Создать производный класс «БУЛЕВА МАТРИЦА» от класса «ЦЕЛОЧИСЛЕННАЯ МАТРИЦА».

Методы: конструктор без параметров; конструктор, принимающий в качестве параметров целочисленный двумерный массив, содержащий матрицу, и ее размеры *n* и *m*; конструктор копирования; деструктор; метод возвращает число единиц в матрице; метод, возвращающий α-каноническую матрицу (в исходной матрице удалены повторяющиеся строки; строки составлены в порядке возрастания неотрицательных чисел, в качестве двоичных кодов которых рассматриваются данные строки).

Переопределить операции: поэлементная конъюнкция двух матриц (&); поэлементная дизъюнкция двух матриц (|); поэлементная операция ИСКЛЮЧАЮЩЕЕ ИЛИ двух матриц (^); произведение двух матриц A=[aij] и B=[ bjk], где i=, j=, k= (\*). При вычислениях операция целочисленного умноже­ния заменяется конъюнкцией, а сложение – дизъюнкцией; присваивание (=).

10. Создать производный класс «МИНОР» для базового класса «МАТРИЦА» размерности *n×m*. Переопределить для производного класса все операции и методы.

11. Расширить возможности стандартного класса Time, чтобы можно было выводить время дня: утро, вечер и т. д.

12. Расширить возможности стандартного класса Date, чтобы можно было выводить время года: зима, лето и т. д.

13. Расширить возможности класса Annotation, чтобы можно было выводить время и дату изменения аннотации.

14. Расширить возможности класса Dictionary, чтобы можно было выводить дату последнего изменения в словаре.

15. Расширить возможности класса File, чтобы можно было выводить время и дату создания файла.

16. Расширить возможности класса Stack, чтобы можно было выводить время последнего сеанса работы со стеком.

17. Определить базовый класс для работы с прямоугольными матрицами, предусмотрев ввод-вывод матриц и выполнение следующих операций: сложение матриц; умножение матрицы на скаляр; перестановка строк матрицы по заданному вектору транспозиции; перестановка столбцов матрицы по заданному вектору транспозиции. В производном классе реализовать указанные операции для квадратных матриц, добавив выполнение следующих операций:транспонирование матрицы; умножение матриц.