

## **Лабораторная работа №7**

### **Наследование классов**

В заданиях требуется описать абстрактный базовый класс и производные от него и создать параметризованную коллекцию объектов производных классов. Обеспечить читабельный вывод полей классов на экран.

Используя механизм виртуальных методов, продемонстрировать единообразную работу с элементами коллекции.

#### **Индивидуальные задания на лабораторную работу**

##### **Вариант 1**

Создать абстрактный класс File, инкапсулирующий в себе методы Open, Close, Seek, Read, Write, GetPosition и GetLength. Создать производные классы MyDataFile1 и MyDataFile2— файлы, содержащие в себе данные некоторого определенного типа MyData1 и MyData2, а также заголовки, облегчающие доступ к этим файлам.

Создать класс Folder, содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода списка имен и длин файлов. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

##### **Вариант 2**

Создать абстрактный класс Point (точка). На его основе создать классы ColoredPoint и Line. На основе класса Line создать класс ColoredLine и класс PolyLine (многоугольник). Все классы должны иметь виртуальные методы установки и получения значений всех координат, а также изменения цвета и получения текущего цвета.

Создать класс Picture, содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

##### **Вариант 3**

Создать абстрактный класс Vehicle. На его основе реализовать классы Car (автомобиль), Bicycle (велосипед) и Lorry (грузовик). Классы должны иметь возможность задавать и получать параметры средств передвижения (цена, максимальная скорость, год выпуска и т.д.). Наряду с общими полями и методами, каждый класс должен содержать и специфичные для него поля.

Создать класс Garage, содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть

возможность вывода характеристик объектов списка. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

#### **Вариант 4**

Создать абстрактный класс Figure. На его основе реализовать классы Rectangle (прямоугольник), Circle (круг) и Trapezium (трапеция) с возможностью вычисления площади, центра тяжести и периметра.

Создать класс Picture, содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

#### **Вариант 5**

Создать абстрактный класс Number с виртуальными методами, реализующими арифметические операции. На его основе реализовать классы Integer и Real.

Создать класс Series (набор), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

#### **Вариант 6**

Создать абстрактный класс Body. На его основе реализовать классы Parallelepiped (прямоугольный параллелепипед), Cone (конус) и Ball (шар) с возможностью вычисления площади поверхности и объема.

Создать класс Series (набор), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

#### **Вариант 7**

Создать абстрактный класс Currency для работы с денежными суммами. Определить в нем методы перевода в рубли и вывода на экран. На его основе реализовать классы Dollar, Euro и Pound (фунт стерлингов) с возможностью пересчета в центы и пенсы соответственно.

Создать класс Purse (кошелек), содержащий массив/параметризованную коллекцию объектов этих классов в

динамической памяти. Предусмотреть возможность вывода общей суммы, переведенной в рубли, и суммы по каждой из валют. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 8**

Создать абстрактный класс Triangle (треугольник), задав в нем длину двух сторон, угол между ними, методы вычисления площади и периметра. На его основе создать классы, описывающие равносторонний, равнобедренный и прямоугольный треугольники со своими методами вычисления площади и периметра.

Создать класс Picture, содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка и получения суммарной площади. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 9**

Создать абстрактный класс Solution (решение) с виртуальными методами вычисления корней уравнения и вывода на экран. На его основе реализовать классы Linear (линейное уравнение) и Square (квадратное уравнение).

Создать класс Series (набор), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 10**

Создать абстрактный класс Function (функция) с виртуальными методами вычисления значения функции  $y = f(x)$  в заданной точке  $x$  и вывода результата на экран. На его основе реализовать классы Ellipse, Hiperbola и Parabola.

Создать класс Series (набор), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 11**

Создать абстрактный класс Triad (тройка) с виртуальными методами

увеличения на 1. На его основе реализовать классы Date (дата) и Time (время).

Создать класс Memories, содержащий массив/параметризованную коллекцию пар (дата-время) объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка и выборки самого раннего и самого позднего событий. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 12**

Описать абстрактный класс Element (элемент логической схемы), задав в нем числовой идентификатор, количество входов, идентификаторы присоединенных к нему элементов (до 10) и двоичные значения на входах и выходе. На его основе реализовать классы AND и OR — двоичные вентили, которые могут иметь различное количество входов и один выход и реализуют логическое умножение и сложение соответственно.

Создать класс Scheme (схема), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможности вывода характеристик объектов списка и вычисление значений, формируемых на выходах схемы по заданным значениям входов. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 13**

Описать абстрактный класс Element (элемент логической схемы) задав в нем символьный идентификатор, количество входов, идентификаторы присоединенных к нему элементов (до 10) и двоичные значения на входах и выходе. На его основе реализовать классы AND\_NOT и OR\_NOT — двоичные вентили, которые могут иметь различное количество входов и один выход и реализуют логическое умножение с отрицанием и сложение с отрицанием соответственно.

Создать класс Scheme (схема), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможности вывода характеристик объектов списка и вычисление значений, формируемых на выходах схемы по заданным значениям входов. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 14**

Описать абстрактный класс Trigger (триггер), задав в нем идентификатор и двоичные значения на входах и выходах. На его основе реализовать классы RS и JK, представляющие собой триггеры

соответствующего типа.

Создать класс Register (регистр), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможности вывода характеристик объектов списка, общего сброса и установки значений каждого триггера по заданным значениям входов. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 15**

Создать абстрактный класс Progression (прогрессия) с виртуальными методами вычисления заданного элемента и суммы прогрессии. На его основе реализовать классы Linear (арифметическая) и Exponential (геометрическая).

Создать класс Series (набор), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка и вывода общей суммы всех прогрессий. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 16**

Создать абстрактный класс Pair (пара значений) с виртуальными методами, реализующими арифметические операции. На его основе реализовать классы Fractional (дробное) и LongLong (длинное целое).

В классе Fractional вещественное число представляется в виде двух целых, в которых хранятся целая и дробная часть числа соответственно. В классе LongLong длинное целое число хранится в двух целых полях в виде старшей и младшей части.

Создать класс Series (набор), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода характеристик объектов списка и вывода общей суммы всех значений. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 17**

Создать абстрактный класс Integer (целое) с символьным идентификатором, виртуальными методами, реализующими арифметические операции, и методом вывода на экран. На его основе реализовать классы Decimal (десятичное) и Binary (двоичное). Число представить в виде массива цифр.

Создать класс Series (набор), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода значений и идентификаторов всех объектов списка и

вывода общей суммы всех десятичных значений. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 18**

Создать абстрактный класс `Sorting` (сортировка) с идентификатором последовательности, виртуальными методами сортировки, получения суммы и вывода на экран. На его основе реализовать классы `Choice` (метод выбора) и `Quick` (быстрая сортировка).

Создать класс `Series` (набор), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода идентификаторов и сумм элементов каждого объекта списка, а также вывода общей суммы всех значений. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 19**

Создать абстрактный класс `Pair` (пара значений) с виртуальными методами, реализующими арифметические операции, и методом вывода на экран. На его основе реализовать классы `Money` (деньги) и `Complex` (комплексное число).

В классе `Money` денежная сумма представляется в виде двух целых, в которых хранятся рубли и копейки соответственно. При выводе части числа снабжаются словами «руб.» и «коп.». В классе `Complex` предусмотреть при выводе символ мнимой части (*i*).

Создать класс `Series` (набор), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода объектов списка. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по некоторому критерию, вывода в файл и считывания из файла.

### **Вариант 20**

Создать абстрактный класс `Worker` с полями, задающими фамилию работника, фамилии руководителя и подчиненных и виртуальными методами вывода списка обязанностей и списка подчиненных на экран. На его основе реализовать классы `Manager` (руководитель проекта), `Developer` (разработчик) и `Coder` (младший программист).

Создать класс `Group` (группа), содержащий массив/параметризованную коллекцию объектов этих классов в динамической памяти. Предусмотреть возможность вывода всех объектов списка и выборки по фамилии с выводом всего дерева подчиненных. Написать демонстрационную программу, в которой будут использоваться все методы классов.

Дополнительное задание: дополнить класс методами сортировки по

некоторому критерию, вывода в файл и считывания из файла.