

## Лабораторная работа №2

Тематика заданий направлена на изучения основ объектно-ориентированного программирования с использованием языка C++.

Для подготовки заданий, использовалась книга «Павловская Т. А., Щупак Ю. А., C++. Объектно-ориентированное программирование. / Т.А, Павловская, Ю.А, Щупак, Спб.: Питер, 2006», доступная в электронном варианте.

**Для успешной защиты лабораторной работы необходимо:** Выполнить задание в полном объеме, к моменту защиты (следующее занятие) иметь распечатанный отчет в котором указывается выполненное задание, приводятся тестовые примеры работы программы с результатами работы и приводятся исходные тексты реализованной программы. Так же в отчете необходимо отразить, что было изучено для выполнения данной лабораторной работы.

### Задание.

Необходимо придерживаться одного из возможных стандартов кодирования и оформления исходных текстов. По возможности (там, где это уместно) реализовать перегруженные операторы для реализованных вами классов.

За лабораторное занятие **обязательно** необходимо выполнить базовую часть задания (классы с основными методами), дома реализовать оставшееся. На следующем занятии осуществляется защита лабораторной работы.

### Вариант 1

Описать класс, реализующий бинарное дерево, обладающее возможностью добавления новых элементов, удаления существующих, поиска элемента по ключу, а также последовательного доступа ко всем элементам.

Написать программу, использующую этот класс для представления англо-русского словаря. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Предусмотреть возможность формирования словаря из файла и с клавиатуры.

### Вариант 2

Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол.

Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

### Вариант 3

Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### Вариант 4

Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с

получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### Вариант 5

Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменение размеров, построение наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### Вариант 6

Составить описание класса многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, операции сложения, вычитания и умножения многочленов с получением нового объекта-многочлена, вывод на экран описания многочлена.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### Вариант 7

Составить описание класса, обеспечивающего представление матрицы произвольного размера с возможностью изменения числа строк и столбцов, вывода на экран подматрицы любого размера и всей матрицы.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### Вариант 8

Описать класс, реализующий тип данных «длинное целое со знаком» и работу с ними. Длина целого числа может быть неограниченной. Класс должен реализовывать следующие операции:

- сложение, вычитание, умножение, деление (+, -, \*, /) (умножение и деление, как на другое длинное целое, так и на целое);
- комбинированные операции присваивания (+=, -=, \*=, /=);
- операции сравнения на равенство/неравенство и больше/меньше;
- операции ввода/вывода в стандартные потоки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### Вариант 9

Описать класс, реализующий тип данных «вещественная матрица» и работу с ними. Класс должен реализовывать следующие операции над матрицами:

- сложение, вычитание, умножение, деление (+, -, \*, /) (умножение и деление, как на другую матрицу, так и на число);
- комбинированные операции присваивания (+=, -=, \*=, /=);
- операции сравнения на равенство/неравенство;
- операции вычисления обратной и транспонированной матрицы, операцию возведения в степень;
- методы вычисления детерминанта и нормы;

- методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная, симметрическая, верхняя треугольная, нижняя треугольная);
- операции ввода/вывода в стандартные потоки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

## Вариант 10

Описать класс «множество», реализующий как ориентированное так и неориентированное множество позволяющий выполнять основные операции — добавление и удаление элемента, пересечение, объединение и разность множеств, .

Класс должен реализовывать следующие операции:

- комбинированные операции присваивания ( $+=$ ,  $-=$ ,  $*=$ ,  $/=$ );
- операции сравнения на равенство/неравенство;
- операции ввода/вывода в стандартные потоки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

## Вариант 11

Описать класс, реализующую игру «Крестики-нолики»(большое поле) между двумя игроками. Реализовать методы для проверки возможности установки крестика/нолика в указанной позиции, для установки крестика/нолика в указанной позиции, для проверки победы одного из игроков.

Класс должен реализовывать следующие операции:

- операции сравнения на равенство/неравенство;
- операции ввода/вывода в стандартные потоки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

## Вариант 12

Описать класс, реализующий игру-головоломку «Игра в 15». Начальное размещение номеров — случайное. Реализовать методы для осуществления перестановки клеток, для проверки правильной расстановки клеток.

Класс должен реализовывать следующие операции:

- операции сравнения на равенство/неравенство;
- операции ввода/вывода в стандартные потоки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

## Вариант 13

Описать класс, реализующий игру-головоломку «Кубик Рубика». Начальное размещение цветов — случайное или загружается из файла. Реализовать методы для осуществления поворотов граней куба, для проверки правильной расстановки цветных клеток.

Класс должен реализовывать следующие операции:

- операции сравнения на равенство/неравенство;
- операции ввода/вывода в стандартные потоки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

## Вариант 14

Описать класс, реализующий «Машину Тьюринга». Начальное состояние ленты и программа МТ задаётся с консоли или загружается из файла. Реализовать методы для

добавления/удаления/просмотра правил МТ. Для задания/изменения значений на ленте, для осуществления шага МТ.

Класс должен реализовывать следующие операции:

- операции сравнения на равенство/неравенство;
- операции ввода/вывода в стандартные потоки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.