

ЛАБОРАТОРНАЯ РАБОТА №3.

НАЗВАНИЕ РАБОТЫ: Наследование и полиморфизм.

ЦЕЛЬ РАБОТЫ: Изучение особенностей реализации в языке C++ механизмов наследования и полиморфизма. Изучение особенностей использования классов-контейнеров в объектно-ориентированном программировании.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:

1. Согласовать с преподавателем задание.
2. Построить иерархию классов, составными частями которой будут сущности, указанные в задании.
3. Предусмотреть не менее трех производных классов (кроме задания с элементами графа).
4. Реализовать полученную иерархию наследования средствами языка C++. Включить в базовый класс указанную в задании виртуальную функцию. Переопределить данную функцию в производных классах.
5. Базовый класс иерархии сделать абстрактным.
6. Создать класс-контейнер, владеющий динамическим списком объектов, являющихся экземплярами производных классов из иерархии наследования. Включить в класс-контейнер функцию, позволяющую добавлять новые элементы в динамический список объектов и функцию, обрабатывающую список объектов с использованием механизмов полиморфизма.
7. В задании полужирным шрифтом выделена возможная ключевая функциональность для класса-контейнера.
8. Все классы из иерархии наследования должны включать в себя конструкторы по умолчанию, инициализирующий и копирующий конструкторы. В класс-контейнер в данной работе достаточно включить конструктор по умолчанию.
9. В функции main осуществить демонстрацию функциональности разработанных классов.
10. Подготовить отчет по лабораторной работе.

КРИТЕРИИ ОЦЕНИВАНИЯ:

1. Задания, в которых отсутствует класс-контейнер, оцениваются из максимума в 8 баллов. В этом случае явление полиморфизма необходимо проиллюстрировать в функции main.
2. Задания, в которых фигурирует класс-контейнер, владеющий динамическим списком объектов производных классов, оцениваются из максимума в 9 баллов.
3. Задания, в которых реализована ключевая функциональность для класса-контейнера, оцениваются из максимума в 10 баллов.

СОДЕРЖАНИЕ ОТЧЕТА:

1. Название работы.
2. Цель работы.
3. Содержание задания.
4. Диаграмма классов – графическое представление иерархии наследования.
5. Интерфейсы разработанных классов – прототипы открытых функций и данные.
6. Исходники реализации классов.
7. Исходник функции main.
8. Скринь результатов работы программы.
9. Выводы по работе.

СПИСОК ЗАДАНИЙ:

1. Базовый класс: **Фигура на плоскости** .
Производные классы: окружность, прямоугольник, **ромб**.
Виртуальная функция: расчет площади фигуры.
Ключевая функциональность класса-контейнера: **работа с классом-контейнером в данном задании не предполагается.**
2. Базовый класс: **Фигура на плоскости** .
Производные классы: окружность, прямоугольник, **треугольник**.
Виртуальная функция: расчет площади фигуры.
Ключевая функциональность класса-контейнера: **работа с классом-контейнером в данном задании не предполагается.**
3. Базовый класс: **Фигура на плоскости** .
Производные классы: окружность, прямоугольник, **квадрат**.
Виртуальная функция: расчет площади фигуры.
Ключевая функциональность класса-контейнера: **работа с классом-контейнером в данном задании не предполагается.**
4. Базовый класс: **Фигура на плоскости** .
Производные классы: окружность, прямоугольник, **трапеция**.
Виртуальная функция: расчет площади фигуры.
Ключевая функциональность класса-контейнера: **работа с классом-контейнером в данном задании не предполагается.**
5. Базовый класс: **Фигура на плоскости** .
Производные классы: окружность, прямоугольник, **параллелограмм**.
Виртуальная функция: расчет площади фигуры.

Ключевая функциональность класса-контейнера: **работа с классом-контейнером в данном задании не предполагается.**

6. Базовый класс: **Фигура в пространстве .**

Производные классы: конкретные фигуры в пространстве.

Виртуальная функция: расчет объема фигуры.

Ключевая функциональность класса-контейнера: **нахождение фигуры с наименьшим объемом.**

7. Базовый класс: **Бинарная логическая функция .**

Производные классы: конкретные бинарные логические функции.

Виртуальная функция: вычисление значения функции для конкретных аргументов.

Ключевая функциональность класса-контейнера: **вычисление значения логического выражения, в которое входят все функции из класса-контейнера в предположении, что между ними стоит операция &.**

8. Базовый класс: **Элемент графа.**

Производные классы: дуга и вершина графа.

Виртуальная функция: вывод информации об элементе графа.

Ключевая функциональность класса-контейнера:

а) получение матрицы смежности,

б) получение матрицы инцидентности,

с) реализация некоторого алгоритма на графе (max 10).

9. Базовый класс: **Элемент ориентированного графа.**

Производные классы: дуга и вершина ориентированного графа.

Виртуальная функция: вывод информации об элементе графа.

Ключевая функциональность класса-контейнера:

а) получение матрицы смежности,

б) получение матрицы инцидентности,

с) реализация некоторого алгоритма на ориентированном графе (max 10)

10. Базовый класс: **Шахматная фигура.**

Производные классы: конкретные шахматные фигуры.

Виртуальная функция: проверка на возможность хода фигуры в указанную позицию на пустой доске.

Ключевая функциональность класса-контейнера: **проверка на возможность хода фигуры на заполненной доске.**

11. Базовый класс: **Банковский вклад.**

Производные классы: банковские вклады конкретного типа.

Виртуальная функция: расчет сложного процента по вкладу за некоторый период.

Ключевая функциональность класса-контейнера: **расчет суммарных обязательств по всем вкладам за некоторый период.**

12. Базовый класс: **Работник с различными формами начисления заработной платы.**

Производные классы: работники с конкретными формами начисления заработной платы.

Виртуальная функция: расчет заработной платы работника за некоторый период.

Ключевая функциональность класса-контейнера: **расчет фонда заработной платы для всех работников за некоторый период.**

13. Базовый класс: **Элемент электрической цепи.**

Производные классы: конкретные элементы электрической цепи.

Виртуальная функция: вывод информации об элементе цепи.

Ключевая функциональность класса-контейнера: **проверка цепи на связность.**

14. Базовый класс: **Функция от одной переменной.**

Производные классы: конкретные функции от одной переменной.

Виртуальная функция: вычисление значения функции для конкретного аргумента.

Ключевая функциональность класса-контейнера: **вычисление суммы значений всех функций, содержащихся в контейнере, вычисленных для конкретного аргумента.**

15. Базовый класс: **Функция от двух переменных.**

Производные классы: конкретные функции от двух переменных.

Виртуальная функция: вычисление значения функции для конкретных аргументов.

Ключевая функциональность класса-контейнера: **вычисление произведения значений всех функций, содержащихся в контейнере, вычисленных для конкретного аргумента.**

16. Базовый класс: **Параметрическая функция от одной переменной.**

Производные классы: конкретные параметрические функции от одной переменной.

Виртуальная функция: вычисление значения функции для конкретного аргумента.

Ключевая функциональность класса-контейнера: **вычисление суммы значений всех функций, содержащихся в контейнере, вычисленных для конкретного аргумента.**

17. Базовый класс: **Параметрическая функция от двух переменных.**

Производные классы: конкретные параметрические функции от двух переменных.

Виртуальная функция: вычисление значения функции для конкретных аргументов.

Ключевая функциональность класса-контейнера: **вычисление произведения значений всех функций, содержащихся в контейнере, вычисленных для конкретного аргумента.**

18. Базовый класс: **Последовательность чисел.**

Производные классы: арифметическая, геометрическая последовательности, числа Фибоначчи.

Виртуальная функция: нахождение суммы первых n членов последовательности.

Ключевая функциональность класса-контейнера: **нахождение значения максимальной суммы первых n членов каждой последовательности.**

19. Базовый класс: **Последовательность чисел.**

Производные классы: арифметическая, геометрическая последовательности, числа Фибоначчи.

Виртуальная функция: нахождение значения n -го члена последовательности.

Ключевая функциональность класса-контейнера: **нахождение минимального значения среди n -х членов каждой последовательностей.**