

Владимирский государственный университет

Методические указания
к лабораторным работам по курсу

Объектно-ориентированное программирование

Направления подготовки:

010200 «Математика и компьютерные науки»

010300 «Фундаментальные информатика и информационные технологии»

010500 «Математическое обеспечение и администрирование информационных систем»

Составители:

Воронова Л.М.

Лексин А.Ю.

Владимир 2012

ОГЛАВЛЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА № 1 АБСТРАКЦИЯ И ИНКАПСУЛЯЦИЯ	3
ЛАБОРАТОРНАЯ РАБОТА № 2 ПЕРЕГРУЗКА ОПЕРАЦИЙ	5
ЛАБОРАТОРНАЯ РАБОТА №3 ОДИНОЧНОЕ НАСЛЕДОВАНИЕ И ПОЛИМОРФИЗМ.....	6
ЛАБОРАТОРНАЯ РАБОТА № 4 ШАБЛОННЫЕ КЛАССЫОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.	
ЛАБОРАТОРНАЯ РАБОТА № 5 МНОЖЕСТВЕННОЕ НАСЛЕДОВАНИЕ	10

Лабораторная работа № 1 АБСТРАКЦИЯ И ИНКАПСУЛЯЦИЯ

Цель работы

Изучить основы объектно-ориентированного программирования на алгоритмическом языке C++. Получить представление о реализации принципов абстракции и инкапсуляции.

Порядок выполнения работы

1. Изучить семантику и синтаксис описания классов на языке C++.
2. Реализовать класс **Complex** для работы с комплексными числами. Класс должен обеспечивать инкапсуляцию информации о действительной и мнимой частях комплексного числа, а также о его модуле и амплитуде (в представлении Эйлера). Соответствующие поля должны являться закрытыми (**private**), а для доступа к их значениям должны быть описаны методы записи и чтения (например, для работы с действительной частью комплексного числа это могут быть методы **setRe()** и **getRe()**). Между двумя представлениями комплексного числа ($a+ib$ и $\rho \exp(i\varphi)$) должна осуществляться синхронизация при любых изменениях значения комплексного числа в том или ином представлении. Конструкторы класса должны обеспечивать возможность создания объекта тремя способами: со значениями полей по умолчанию (действительная и мнимая части равны нулю), создания объекта с заданием действительной и мнимой частей, а также создания объекта в представлении Эйлера. Для класса должны быть реализованы арифметические методы (сложение, вычитание, умножение, деление) и специальные методы ввода/вывода.
3. Разработать и реализовать тестовое приложение для класса **Complex**, демонстрирующее использование его методов.
4. Выполнение работы продемонстрировать преподавателю.
5. После приёмки работы преподавателем оформить отчёт.
6. Защитить работу преподавателю, ответив на заданные вопросы.

Содержание отчета

1. Цель работы.
2. Формулировка задания.
3. Текст программы.
4. Диаграмма классов.
5. Тестовые примеры.
6. Выводы по работе, содержащие размеры программы, т.е. количество операторов (строк) в исходном модуле и объем exe-файла.

Контрольные вопросы

1. Понятия инкапсуляции, полиморфизма, наследования.
2. Определение класса.

3. Определение объекта.
4. Поля и методы класса.
5. Конструктор и деструктор класса, их назначение и вызов.
6. Ограничение доступа к полям и методам класса.
7. Указатель `this`. Назначение и использование.
8. Оператор разрешения области видимости `::`.
9. Динамическое распределение памяти под объекты и массивы объектов.
10. Типизация при присваивании объектов и указателей на объекты.

Лабораторная работа № 2 ПЕРЕГРУЗКА ОПЕРАЦИЙ

Цель работы

Изучить механизмы перегрузки операций в языке C++.

Порядок выполнения работы

1. Изучить семантику и синтаксис перегрузки операций в языке C++.
2. Модифицировать класс **Complex**, созданный в лабораторной работе №1.
 - 2.1. Описать конструктор инициализации (копирования)
Complex::Complex(Complex&)
 - 2.2. Вместо арифметических методов перегрузить операции +, -, *, /.
При этом первые две из них реализовать как методы класса, а операции умножения и деления реализовать как дружественные функции.
 - 2.3. Перегрузить операцию =.
 - 2.4. Вместо методов ввода/вывода реализовать перегрузку операций >> и <<.
3. Разработать и реализовать тестовое приложение для модифицированного класса **Complex**, демонстрирующее различные варианты использования перегруженных операций.
4. Выполнение работы продемонстрировать преподавателю.
5. После приёма работы преподавателем оформить отчёт.
6. Защитить работу преподавателю, ответив на заданные вопросы.

Содержание отчета

1. Цель работы.
2. Формулировка задания.
3. Текст программы.
4. Диаграмма классов.
5. Тестовые примеры.
6. Выводы по работе, содержащие размеры программы, т.е. количество операторов (строк) в исходном модуле и объем exe-файла.

Контрольные вопросы

1. Перегрузка операторов в виде утилит.
2. Операторы-методы классов.
3. Ограничения на перегрузку операторов.
4. Перегрузка операторов потокового ввода/вывода.
5. Конструктор инициализации.

Лабораторная работа №3 ОДИНОЧНОЕ НАСЛЕДОВАНИЕ И ПОЛИМОРФИЗМ

Цель работы

Изучить реализацию одиночного наследования и полиморфизма в языке C++.

Порядок выполнения работы

1. Изучить семантику и синтаксис создания производных классов на языке C++, а также способы создания полиморфических кластеров.
2. На основе класса **Complex**, реализованного в лабораторной работе №2, описать производный класс **Vector**, представляющий вектор в двумерном Евклидовом пространстве.
 - 2.1. Описать необходимые методы и дружественные функции для класса **Vector**.
 - 2.2. Реализовать виртуальную функцию **print()**, выводящую значение объекта в формате, соответствующем смыслу этого объекта ($a+ib$ для комплексного числа и $(x;y)$ для вектора).
 - 2.3. Реализовать необходимые способы приведения типов между классами **Complex** и **Vector**.
3. Разработать и реализовать тестовое приложение для классов **Complex** и **Vector**, демонстрирующее использование виртуальной функции и различные варианты приведения типов.
4. Выполнение работы продемонстрировать преподавателю.
5. После приёмки работы преподавателем оформить отчёт.
6. Защитить работу преподавателю, ответив на заданные вопросы.

Содержание отчета

1. Цель работы.
2. Формулировка задания.
3. Текст программы.
4. Диаграмма классов.
5. Тестовые примеры.
6. Выводы по работе, содержащие размеры программы, т.е. количество операторов (строк) в исходном модуле и объем exe-файла.

Контрольные вопросы

1. Виртуальные методы и позднее связывание.
2. Таблица виртуальных методов.
3. Механизм вызова виртуальных методов.
4. Чистая виртуальная функция.
5. Раннее связывание методов и объектов.
6. Механизм вызова виртуальных функций.
7. Наследование классов.

Лабораторная работа № 4 ШАБЛОННЫЕ КЛАССЫ

Цель работы

Изучить средства обобщения классов в языке C++.

Порядок выполнения работы

1. Изучить семантику и синтаксис создания шаблонных (обобщённых) классов на языке C++.
2. Разработать обобщенный класс по заданию преподавателя. В нем реализовать методы: ввод информации о параметрах и элементах класса; вывод элементов на дисплей; методы индивидуального задания.
3. Составить головную функцию, в которой осуществить создание объектов конкретных классов путем параметризации разработанного обобщенного класса простыми типами данных (действительный, целый).
4. Выполнить все предусмотренные методы полученных классов.
5. Разработать тестовые наборы входных данных. Провести тестирование.
6. Выполнение работы продемонстрировать преподавателю.
7. После приёмки работы преподавателем оформить отчёт.
8. Защитить работу преподавателю, ответив на заданные вопросы.

Содержание отчета

1. Цель работы.
2. Формулировка задания.
3. Текст программы.
4. Диаграмма классов.
5. Тестовые примеры.
6. Выводы по работе, содержащие размеры программы, т.е. количество операторов (строк) в исходном модуле и объем exe-файла.

Варианты заданий

1. Класс «Квадратная матрица». Методы: умножение на матрицу, умножение на константу, вывод элементов матрицы на дисплей.
2. Класс «Квадратная матрица». Методы: сложение с матрицей, транспонирование, вывод элементов матрицы на дисплей.
3. Класс «Квадратная матрица». Методы: возведение матрицы в квадрат, вычисление следа матрицы, сравнение матриц на равенство, вывод элементов матрицы на дисплей.
4. Класс «Массив». Методы: сортировка массива в порядке возрастания, перестроение элементов обратном порядке, вывод элементов массива на дисплей.
5. Класс «Массив». Методы: объединение двух массивов, перестроение порядка следования элементов на противоположный, вывод элементов массива на дисплей.

6. Класс «Массив». Методы: получение минимального значения из элементов массива, получение максимального значения из элементов массива, поэлементное сложение двух массивов, отображение элементов массива на дисплее.

7. Класс «Вектор». Методы: скалярное умножение векторов, вычисление длины вектора, сложение векторов, умножение вектора на константу, отображение всех компонент вектора на дисплее.

8. Класс «Очередь». Методы: добавление элемента, удаление элемента, удаление из очереди всех элементов, равных заданному значению.

9. Класс «Дек» (двусторонняя очередь). Методы: добавление и удаление элементов. Примечание: реализовать на базе связного списка.

10. Класс «Множество без повторений». Методы: включение элемента, исключение элемента, проверка принадлежности элемента множеству, объединение множеств, отображение элементов множества на дисплее.

11. Класс «Множество с повторениями». Методы: включение элемента, исключение элемента, определение количества вхождений заданного значения во множество, объединение множеств, отображение всех элементов множества на дисплее.

12. Класс «Словарь без повторений». Методы: добавление элементов словаря, удаление элементов словаря по ключу, получение информационного значения по ключу, отображение элементов словаря на дисплее. Примечание: в данном случае следует использовать параметризацию обобщенного класса двумя типами.

13. Класс «Словарь с повторениями». Методы: добавление элементов, исключение элементов, вывод на дисплей элементов словаря с заданным ключом, вывод на дисплей всех элементов словаря. Примечание: в данном случае следует использовать параметризацию обобщенного класса двумя типами.

14. Класс «Односвязный список». Методы: добавление элемента в список, удаление элемента из списка, удаление всех элементов списка, определение порядкового номера элемента по заданному значению, отображение всех элементов списка на дисплее.

15. Класс «Циклический список». Методы: добавление элемента в список, удаление элемента из списка, отображение всех элементов списка на дисплее.

16. Класс «Квадратная матрица». Методы: вычисление суммы положительных элементов каждой строки и запись ее в массив, умножение на константу, вывод элементов матрицы на дисплей.

17. Класс «Квадратная матрица». Методы: сложение с матрицей, вычисление произведения положительных элементов каждого столбца и запись его в массив, вывод элементов матрицы на дисплей.

18. Класс «Квадратная матрица». Методы: вычисление суммы отрицательных элементов каждого столбца и запись его в массив, возведение матрицы в квадрат, вычисление следа матрицы, сравнение матриц на равенство, вывод элементов матрицы на дисплей.

19. Класс «Массив». Методы: сортировка массива в порядке убывания, перестроение элементов обратном порядке, вывод элементов массива на дисплей.

20. Класс «Квадратная матрица». Методы: вычисление следа матрицы, сравнение матриц на равенство, получение максимальных элементов каждого столбца и запись их в массив, вывод элементов матрицы на дисплей.

21. Класс «Массив». Методы: получение максимального и минимального элементов массива и их номеров (индексов), перестроение элементов обратном порядке, вывод элементов массива на дисплей.

Контрольные вопросы

1. Шаблон функции. Сигнатура функции.
2. Отношение обобщения (наполнения) между классами.
3. Последовательность трансляции при параметризации класса.
4. Эквивалентность параметризованных классов.

Лабораторная работа № 5 МНОЖЕСТВЕННОЕ НАСЛЕДОВАНИЕ

Цель работы

Изучить реализацию множественного наследования в языке C++.

Порядок выполнения работы

1. Изучить семантику и синтаксис создания производных классов на языке C++ в случае множественного наследования, а также особенности создания полиморфических кластеров.
2. Описать абстрактный виртуальный базовый класс **Figure**, являющийся базовым для классов, реализующих конкретные геометрические фигуры. В качестве абстрактной функции описать функцию **Show()**, предназначенную для отображения фигуры. Класс должен иметь поля **lineColor**, **bgColor**, соответствующие цвету контура фигуры и цвету заливки. Могут быть также описаны другие поля по усмотрению студента.
3. Описать классы, являющиеся наследниками класса **Figure: Point** (точка), **Circle** (окружность), **Rect** (прямоугольник), с соответствующими наборами свойств и реализацией необходимых методов, в том числе метода **Show()**.
4. Описать класс **RoundRect** (прямоугольник со скруглёнными углами), являющийся наследником классов **Circle** и **Rect**. Определить для него необходимые поля и методы.
5. Разработать и реализовать тестовое приложение с оконным интерфейсом, демонстрирующее работу с описанными классами, включая отображение геометрических фигур на экране. Для хранения набора геометрических фигур использовать массив указателей на класс **Figure**.
6. Выполнение работы продемонстрировать преподавателю.
7. После приёма работы преподавателем оформить отчёт.
8. Защитить работу преподавателю, ответив на заданные вопросы.

Содержание отчета

1. Цель работы.
2. Формулировка задания.
3. Текст программы.
4. Диаграмма классов.
5. Тестовые примеры.
6. Выводы по работе, содержащие размеры программы, т.е. количество операторов (строк) в исходном модуле и объем exe-файла.

Контрольные вопросы

1. Множественное наследование.

2. Проблема неоднозначности наименования полей и методов при множественном наследовании.
3. Проблема повторного наследования.
4. Виртуальные базовые классы.
5. Порядок создания объектов при множественном наследовании классов.
6. Структура объекта при повторном наследовании в случаях использования и не использования механизма виртуальных базовых классов.