МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ОСНОВЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Методические рекомендации к лабораторным работам для студентов специальности 1-28 01 02 «Электронный маркетинг» очной и заочной форм обучения

Могилев 2020

УДК 621.01
ББК 36.4
О87

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления» «13» ноября 2020 г., протокол № 5

Составители: ст. преподаватель Ю. В. Вайнилович;
ст. преподаватель Л. А. Тимашкова

Рецензент Ю. С. Романович

Методические рекомендации к лабораторным работам предназначены для студентов специальности 1-28 01 02 «Электронный маркетинг» очной и заочной форм обучения.

Учебно-методическое издание

ОСНОВЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Ответственный за выпуск	А. И. Якимов
Корректор	Т. А. Рыжикова
Компьютерная верстка	Е. В. Ковалевская

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 26 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, Могилев.

© Белорусско-Российский
университет, 2020

Содержание

Введение…… 4
1 Лабораторная работа № 1. Классы, объекты, организация ввода/вывода, операторы динамического выделения и освобождения памяти……. 5
2 Лабораторная работа № 2. Программирование алгоритмов с использованием конструкторов. ……..... 7
3 Лабораторная работа № 3. Программирование алгоритмов с использованием механизма перегрузки операторов …… 8
4 Лабораторная работа № 4. Программирование алгоритмов с использованием механизма перегрузки функций …… 11
5 Лабораторная работа № 5. Программирование алгоритмов с использованием механизма простого наследования …… 12
6 Лабораторная работа № 6. Реализация принципа полиморфизма. Виртуальные функции. Абстрактные классы …… 13
7 Лабораторная работа № 7. Программирование алгоритмов с использованием механизма множественного наследования …… 15
8 Лабораторная работа № 8. Использование свойств наследования для расширения и уточнения классов …….. 18
9 Лабораторная работа № 9. Дружественные функции и классы..…… 18
10 Лабораторная работа № 10. Шаблоны …… 20
11 Лабораторная работа № 11. Организация работы с текстовыми файлами ……. 24
12 Лабораторная работа № 12. Организация работы с бинарными файлами…….. 26
13 Лабораторная работа № 13. Исключения. Реализация механизма обработки исключений при разработке программ ……. 27
14 Лабораторная работа № 14. Контейнерные классы. ……. 28
15 Лабораторная работа № 15. Разработка собственного контейнерного класса по аналогии с классами STL …….. 31
16 Лабораторная работа № 16. Разработка приложений с использованием паттернов проектирования ……... 35
Список литературы……. 37

Введение

При изучении дисциплины «Основы объектно-ориентированного программирования» студенты выполняют лабораторные работы, варианты которых приведены в методических рекомендациях.

Каждая лабораторная работа соответствует темам лекций и содержит в себе пятнадцать вариантов индивидуальных заданий.

Варианты заданий выдаются студентам заранее с тем, чтобы они имели возможность подготовиться к выполнению лабораторной работы: просмотреть теоретический материал по теме работы и продумать алгоритмы решения задач.

Программы пишутся на языке Си++. Каждую программу в работающем виде (после отладки и тестирования) студент показывает преподавателю, после чего лабораторная работа подлежит защите.

К защите работы студент подготавливает отчет, включающий в себя титульный лист, формулировку задания, описание исходных, результирующих данных и вспомогательных переменных, алгоритм решения задачи, текст программы и результаты ее тестирования.

Защита лабораторной работы состоит из двух частей: практической и теоретической. В практической части студент объясняет принципы работы представленной им программы, в теоретической – отвечает на вопросы по теме лабораторной работы.

При подготовке к защите студенту рекомендуется ответить на контрольные вопросы.

1 Лабораторная работа № 1. Классы, объекты, организация ввода/вывода, операторы динамического выделения и освобождения памяти

Цель работы: получение навыков в разработке программ с использованием классов.

Теоретические сведения

1 Описание классов [1, с. 179–180].
2 Указатель this [1, с. 180–181].

Задания для самостоятельного выполнения

1 Описать класс, реализующий стек. Написать программу, использующую этот класс для моделирования Т-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.

2 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.

3 Создать класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long – для рублей и типа unsigned char – для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения.

4 Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса.

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

5 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.

6 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.

7 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.

8 Составить описание класса для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы массива, возможность задания произвольных границ индексов при создании объекта, возможность выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, вывода на экран элемента массива по заданному индексу, вывода на экран всего массива.

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

9 Составить описание класса для определения одномерных массивов строк фиксированной длины. Предусмотреть возможность обращения к отдельным строкам массива по индексам, контроль выхода за пределы массива, выполнения операций поэлементного сцепления двух массивов с образованием нового массива, слияния двух массивов с исключением повторяющихся элементов, вывод на экран элемента массива по заданному индексу и всего массива.

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

10 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.

Контрольные вопросы

1 Дайте определение терминам «класс» и «объект». Как соотносятся эти понятия между собой?

2 Приведите синтаксис создания объекта в общем виде. Проиллюстрируйте его фрагментом программы.

3 Какие члены класса содержат код?

4 Какие члены класса содержат данные?

5 Перечислите модификаторы доступа к членам класса.

6 Объясните принцип инкапсуляции и его применение к классам.

2 Лабораторная работа № 2. Программирование алгоритмов с использованием конструкторов

Цель работы: получение навыков в разработке программ с использованием конструкторов класса.

Теоретические сведения

1 Указатель this [1, с. 180–181].

2 Конструкторы [1, с. 182–184].

3 Конструктор копирования [1, с. 184–185].

Задания для самостоятельного выполнения

1 Для класса, разработанного в лабораторной работе № 1, определить конструкторы без параметров, с параметрами, копирования.

2 Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор вызвал).

Контрольные вопросы

1 В чём состоит назначение конструктора?

2 Чем конструктор отличается от обычного метода?

3 Перечислите типы конструкторов класса.

4 Сколько конструкторов может содержать класс?

5 Объясните механизм вызова перегружаемого конструктора с помощью ключевого слова this.

3 Лабораторная работа № 3. Программирование алгоритмов с использованием механизма перегрузки операторов

Цель работы: получение навыков в разработке программ с использованием перегрузки операторов.

Теоретические сведения

Перегрузка операций [1, с. 189–197].

Задания для самостоятельного выполнения

1 Определить класс Vec4 как вектор из четырех float. Определить оператор operator[] для Vec4. Определить операторы +, –, *, /, =, +=, –=, *= и /= для комбинаций векторов и чисел с плавающей точкой.

Написать программу, демонстрирующую работу с этим классом.

2 Используя перегрузку функций и перегрузку операторов, создать класс String. Строку представить как массив символов с размещением в динамической памяти. Изучить функции обработки строк библиотеки языка С и реализовать каждую из этих функций как элементы класса String путем перегрузки операторов. Использовать затем эти функции для выполнения операций с текстами.

3 Определить класс RINT, который ведет себя как int, за исключением того, что допустимы только операторы +, – (унарный и бинарный), *, /, %. Подсказка: не определять RINT :: operator int ().

Написать программу, демонстрирующую работу с этим классом.

4 Определить класс Matr4 как вектор, состоящий из четырех Vec4 (см. вариант 1). Определить operator[ ](), возвращающий Vec4 из Matr4. Определить операции транспонирования матрицы, умножения матрицы на число, сложения матриц и умножения матриц.

Написать программу, демонстрирующую работу с этим классом.

5 Создать класс Point для работы с точками на плоскости. Координаты точки – декартовы. Обязательно должны быть реализованы перемещение точки по оси X, перемещение по оси Y, определение расстояния до начала координат, расстояния между двумя точками, преобразование в полярные координаты, сравнение на совпадение и несовпадение.

Написать программу, демонстрирующую работу с этим классом.

6 Создать класс Time для работы со временем в формате «час:минута:секунда». Класс должен включать в себя не менее четырех функций инициализации: числами, строкой (например, «23:59:59»), секундами и временем. Обязательными операциями являются вычисление разницы между двумя моментами времени в секундах, сложение времени и заданного количества секунд, вычитание из времени заданного количества секунд, сравнение моментов времени, перевод в секунды, перевод в минуты (с округлением до целой минуты).

Написать программу, демонстрирующую работу с этим классом.

7 Описать класс Matrix, реализующий тип данных «вещественная матрица» и работу с ними. Класс должен реализовывать путем перегрузки операторов 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.

8 Разработать класс Set (множество). Внутренним представлением класса Set является массив элементов множества (массив располагается в динамической памяти). Класс должен обеспечивать следующие возможности:

– добавление элементов в множество с исключением дублирования элементов;

– удаление элементов из множества;

– вывод множества на экран;

– выполнение операций пересечения, объединения и разности множеств;

– выполнение операций сравнения множеств.

Написать программу, демонстрирующую работу с этим классом.

9 Разработать класс Complex для представления комплексных чисел. Этот класс позволяет осуществлять операции над комплексными числами. Они представляются в виде Re + Im*i, где i имеет значение корня квадратного из –1. С помощью этого класса должны выполняться следующие операции над комплексными числами:

– ввод и вывод комплексного числа посредством перегрузки операций >> и <<;

– присваивание значения одного комплексного числа другому путем перегрузки операции =;

– сложение, вычитание и умножение комплексных чисел путем перегрузки соответственно операций +, – и *;

– сравнение двух комплексных чисел путем перегрузки операций <, >, ==.

Написать программу, демонстрирующую работу с этим классом.

10 Разработать класс Polinomial (полином). Внутренним представлением класса Polinomial является массив членов полинома (массив располагается в динамической памяти). Каждый член содержит коэффициент и показатель степени. Член $2x^4$ имеет коэффициент 2 и показатель степени 4. Разработать полный класс, содержащий соответствующие функции конструктора, конструктора копий, деструктора, а также функцию инициализации set() и функцию вывода на экран get(). Класс должен обеспечивать путем использования перегруженных операций следующие возможности:

– перегрузить операцию сложения (+), чтобы складывать два объекта класса Polinomial;

– перегрузить операцию вычитания (–), чтобы вычитать два объекта класса Polinomial;

– перегрузить операцию присваивания (=), чтобы присваивать один объект класса Polinomial другому;

– перегрузить операцию умножения, чтобы перемножать два объекта класса Polinomial;

– перегрузить операции сложения с присваиванием (+=), вычитания с присваиванием (–=), операцию умножения с присваиванием (*=).

Написать программу, демонстрирующую работу с этим классом.

Контрольные вопросы

1 Каково назначение перегрузки операторов?

2 Как используется ключевое слово operator?

3 Можно ли перегрузкой отменить очередность выполнения операции?

4 Истинно ли следующее утверждение: операция >= может быть перегружена?

5 Сколько аргументов требуется для определения перегруженной унарной операции?

6 Если перегрузить операцию арифметического присваивания, куда передастся результат?

7 Истинно ли следующее утверждение: выражение *objA = objB* будет причиной ошибки компилятора, если объекты разных типов?

8 Можно ли перегрузкой отменить число операндов?

9 Какие операции требуют, чтобы левый операнд был объектом класса?

10 Какие операторы нельзя перегружать?

4 Лабораторная работа № 4. Программирование алгоритмов с использованием механизма перегрузки функций

Цель работы: получение навыков в разработке программ с использованием перегрузки функций.

Теоретические сведения

Перегрузка функций [1, с. 83–85].

Задания для самостоятельного выполнения

Выполнить перегрузку функции для следующих типов параметров:
- одномерный массив типа int размерностью N;
- одномерный массив типа float размерностью N;
- одномерный массив типа double размерностью N.

Варианты

Определить функцию, находящую:
1) максимальный отрицательный элемент;
2) среднеарифметическое отрицательных элементов массива;
3) минимальный положительный элемент;
4) сумму положительных элементов массива;
5) среднеарифметическое положительных элементов массива;
6) произведение отрицательных элементов массива;
7) среднеарифметическое элементов массива, принадлежащих диапазону [*a, b*];
8) произведение элементов массива, принадлежащих диапазону [*a, b*];
9) сумму отрицательных элементов массива;
10) произведение ненулевых элементов массива.

Контрольные вопросы

1 Дайте определение термина «перегрузка конструкторов».
2 Каким образом реализуется перегрузка конструкторов?
3 Что такое копирующий конструктор (copy constructor)?
4 Опишите синтаксис конструктора копирования.
5 Перечислите случаи, когда новый объект следует создавать путём копирования существующего.

5 Лабораторная работа № 5. Программирование алгоритмов с использованием механизма простого наследования

Цель работы: получение навыков использования механизмов наследования и полиморфизма классов. Изучение отношения между классами. Получение навыков построения диаграммы классов на языке UML.

Теоретические сведения

Простое наследование [1, с. 201–205].

Задания для самостоятельного выполнения

1 Создать базовый класс Car (машина), характеризуемый торговой маркой (строка), числом цилиндров, мощностью. Определить методы переназначения и изменения мощности. Создать производный класс Lorry (грузовик), характеризуемый также грузоподъемностью кузова. Определить функции переназначения марки и изменения грузоподъемности.

2 Создать класс Pair (пара чисел); определить методы изменения полей и сравнения пар: пара p1 больше пары p2, если (first.p1 > first.p2) или (first.p1 = first.p2) и (second.p1 > second.p2). Определить класс-наследник Fraction с нолями: целая часть числа и дробная часть числа. Определить полный набор методов сравнения.

3 Создать класс Liquid (жидкость), имеющий поля названия и плотности. Определить методы переназначения и изменения плотности. Создать производный класс Alcohol (спирт), имеющий крепость. Определить методы переназначения и изменения крепости.

4 Создать класс Pair (пара чисел); определить методы изменения полей и вычисления произведения чисел. Определить производный класс Rectangle (прямоугольник) с полями-сторонами. Определить методы вычисления периметра и площади прямоугольника.

5 Создать класс Man (человек) с полями: имя, возраст, пол и вес. Определить методы переназначения имени, изменения возраста и изменения веса. Создать производный класс Student, имеющий поле года обучения. Определить методы переназначения и увеличения года обучения.

6 Создать класс Triad (тройка чисел); определить методы изменения полей и вычисления суммы чисел. Определить производный класс Triangle с полями-сторонами. Определить методы вычисления углов и площади треугольника.

7 Создать класс Triangle с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс Equilateral (равносторонний), имеющий поле площади. Определить метод вычисления площади.

8 Создать класс Triangle с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс RightAngled (прямоугольный), имеющий поле площади. Определить метод вычисления площади.

9 Создать класс Pair (пара чисел); определить методы изменения полей и вычисления произведения чисел. Определить производный класс RightAngled с полями-катетами. Определить методы вычисления гипотенузы и площади треугольника.

10 Создать класс Triad (тройка чисел); определить метод сравнения триад (см. задание 2). Определить производный класс Date с полями: год, месяц и день. Определить полный набор методов сравнения дат.

Контрольные вопросы

1 Что такое производный и базовый классы?

2 В чем заключена основная задача наследования?

3 Пусть базовый класс содержит метод basefunc(), а производный класс не имеет метода с таким именем. Может ли объект производного класса иметь доступ к методу basefunc()? Если да, то при каких условиях?

4 Напишите первую строку описания класса B, который является public-производным класса A.

5 Допустим, что базовый и производный классы включают в себя методы с одинаковыми именами. Какой из методов будет вызван объектом производного класса, если не использована операция разрешения имени?

6 Напишите объявление конструктора без аргументов для производного класса B, который будет вызывать конструктор без аргументов класса A.

6 Лабораторная работа № 6. Реализация принципа полиморфизма. Виртуальные функции. Абстрактные классы

Цель работы: получение навыков в разработке программ с использованием виртуальных функций и абстрактных классов.

Теоретические сведения

1 Виртуальные методы [1, с. 205–207].

2 Абстрактные классы [1, с. 208].

Задания для самостоятельного выполнения

1 Создать абстрактный базовый класс Figure с виртуальными методами вычисления площади и периметра. Создать производные классы Rectangle (прямоугольник), Circle (круг), Trapezium (трапеция) со своими функциями площади и периметра. Самостоятельно определить, какие поля необходимы,

какие из них можно задать в базовом классе, а какие – в производных. Площадь трапеции $S = (a + b) \cdot h / 2$.

2 Создать абстрактный базовый класс Number с виртуальными методами – арифметическими операциями. Создать производные классы Integer (целое) и Real (действительное).

3 Создать абстрактный базовый класс Body (тело) с виртуальными функциями вычисления площади поверхности и объема. Создать производные классы Parallelepiped (параллелепипед) и Ball (шар) со своими функциями площади поверхности и объема.

4 Создать абстрактный класс Currency (валюта) для работы с денежными суммами. Определить виртуальные функции перевода в рубли и вывода на экран. Реализовать производные классы Dollar (доллар) и Euro (евро) со своими функциями перевода и вывода на экран.

5 Создать абстрактный базовый класс Triangle для представления треугольника с виртуальными функциями вычисления площади и периметра. Поля данных должны включать две стороны и угол между ними. Определить классы-наследники: прямоугольный треугольник, равнобедренный треугольник, равносторонний треугольник со своими функциями вычисления площади и периметра.

6 Создать абстрактный базовый класс Root (корень) с виртуальными методами вычисления корней и вывода результата на экран. Определить производные классы Linear (линейное уравнение) и Square (квадратное уравнение) с собственными методами вычисления корней и вывода на экран.

7 Создать абстрактный базовый класс Function (функция) с виртуальными методами вычисления значения функции $y = f(x)$ в заданной точке x и вывода результата на экран. Определить производные классы Ellipse (эллипс), Hyperbola (гипербола) с собственными функциями вычисления y в зависимости от входного параметра x. Уравнение эллипса $x^2/a^2 + y^2/b^2 = 1$; гиперболы $x^2/a^2 - y^2/b^2 = 1$.

8 Создать абстрактный базовый класс Integer (целое) с виртуальными арифметическими операциями и функцией вывода на экран. Определить производные классы Decimal (десятичное) и Binary (двоичное), реализующие собственные арифметические операции и функцию вывода на экран. Число представляется массивом, каждый элемент которого – цифра.

9 Создать абстрактный базовый класс Series (прогрессия) с виртуальными функциями вычисления j-го элемента прогрессии и суммы прогрессии. Определить производные классы Linear (арифметическая) и Exponential (геометрическая). Арифметическая прогрессия $a_j = a_0 + jd$, $j = 0, 1, 2, ...$ Сумма арифметической прогрессии $s_n = (n+1)\ (a_0 + a_n) / 2$. Геометрическая прогрессия $a_j = a_0 r^j$, $j = 0, 1, 2, ...$ Сумма геометрической прогрессии: $s_n = (a_0 - a_n r) / (1 - r)$.

10 Создать абстрактный класс Norm с виртуальной функцией вычисления нормы и модуля. Определить производные классы Complex, Vector3D с собственными функциями вычисления нормы и модуля. Модуль для комплексного числа вычисляется как корень из суммы квадратов действительной и мнимой

частей; норма для комплексных чисел вычисляется как модуль в квадрате. Модуль вектора вычисляется как корень квадратный из суммы квадратов координат; норма вектора вычисляется как максимальное из абсолютных значений координат.

Контрольные вопросы

1 Что такое чистая виртуальная функция? Приведите пример.

2 Напишите объявление чистой виртуальной функции Fun, не возвращающей значений и не имеющей аргументов.

3 Что такое абстрактный класс? Приведите пример.

4 Можно ли создать объект абстрактного класса?

5 Напишите объявление виртуальной функции Fun(), возвращающей результат типа int и имеющей аргумент типа int.

6 Дополните фразу «Пусть указатель p ссылается на объекты базового класса и содержит адрес объекта порожденного класса. Пусть в обоих этих классах имеется виртуальный метод Fun(). Тогда выражение p ->Fun(); поставит на выполнение версию функции Fun() из … класса».

7 Лабораторная работа № 7. Программирование алгоритмов с использованием механизма множественного наследования

Цель работы: получение навыков в разработке программ с использованием механизма множественного наследования.

Теоретические сведения

Множественное наследование [1, с. 208–209].

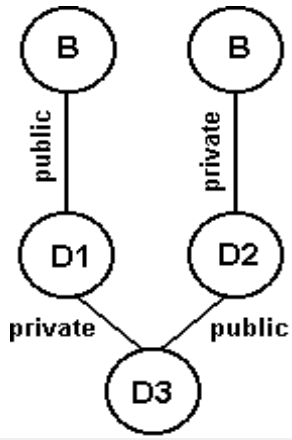
Задания для самостоятельного выполнения

Необходимо построить иерархию классов согласно схеме наследования, приведенной на рисунке 1.

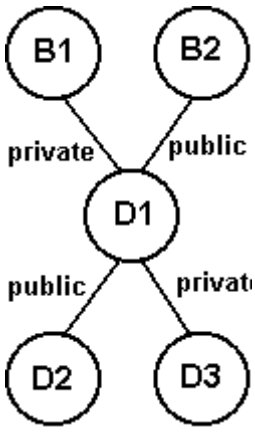
Каждый класс должен содержать инициализирующий конструктор и функцию show для вывода значений.

Функция main должна иллюстрировать иерархию наследования.

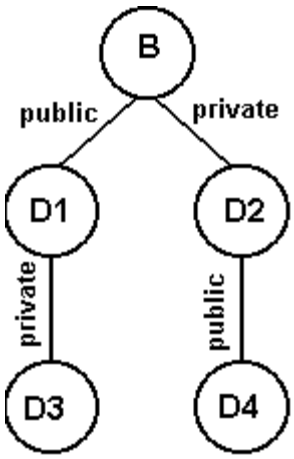
1



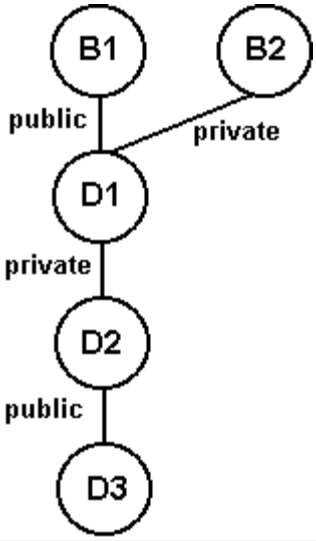
4



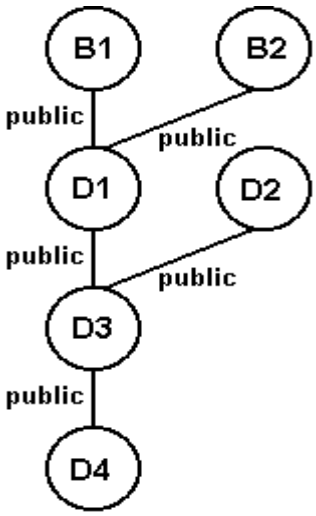
2



5



3



6

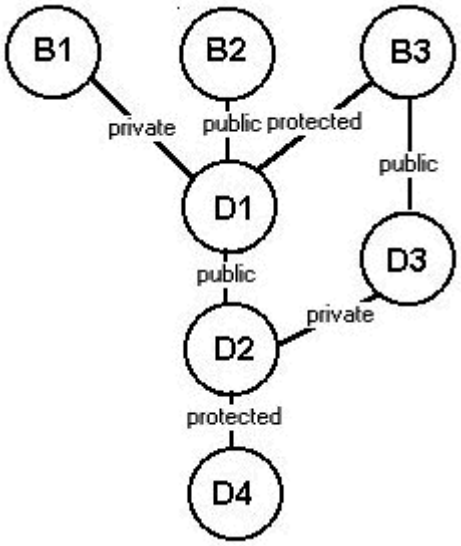
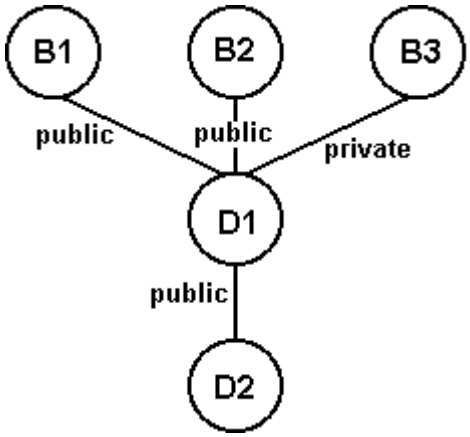
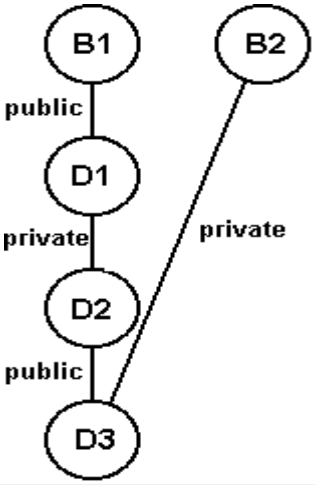


Рисунок 1 – Варианты заданий для самостоятельного выполнения

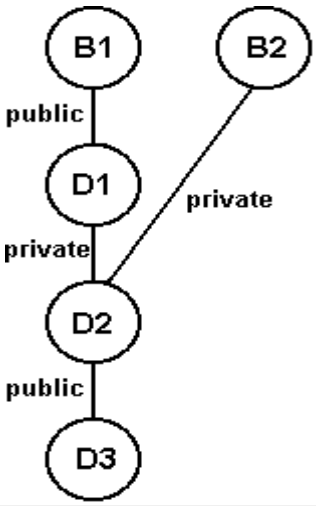
7



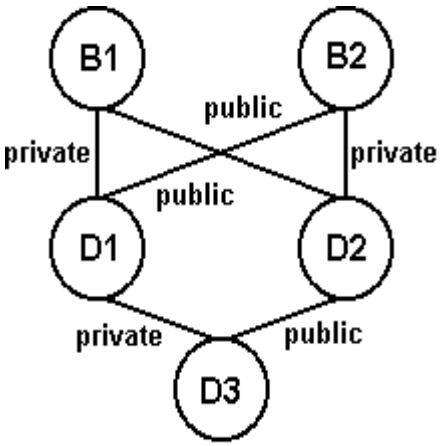
9



8



10



Окончание рисунка 1

Контрольные вопросы

1 Предположим, что существует класс D, производный от базового класса B. Напишите объявление конструктора производного класса, принимающего один аргумент и передающего его в конструктор базового класса.

2 Истинно ли следующее утверждение: класс D может быть производным класса C, который, в свою очередь, является производным класса B, производного от класса A?

3 Напишите первую строку описания класса Petrov, который является public-производным классов Homo и Worker.

4 Дополните фразу «C++ обеспечивает …, которое позволяет производному классу наследовать несколько базовых классов, даже если эти базовые классы неродственные».

5 Истинно ли утверждение о том, что указатель на базовый класс может ссылаться на объекты порожденного класса?

6 Можно ли использовать объект базового класса в производном классе в явном виде?

8 Лабораторная работа № 8. Использование свойств наследования для расширения и уточнения классов

Цель работы: получение навыков моделирования и реализации иерархии классов, а также использования статических данных и методов класса.

Теоретические сведения

Наследование [1, с. 200–211].

Задания для самостоятельного выполнения

1 Определить иерархию классов (в соответствии с вариантом). Базовый класс должен включать обязательный атрибут char *name.

2 Написать демонстрационную программу, в которой создаются объекты различных классов и, по требованию, вызывается метод Show() первого элемента с выводом результата на экран.

Варианты

Иерархии классов:

1) птица, курица, сокол, воробей;
2) служащий, менеджер, рабочий, инженер;
3) деталь, механизм, изделие, узел;
4) журнал, книга, печатное издание, учебник;
5) тест, экзамен, выпускной экзамен, испытание;
6) паспорт, документ, пропуск, удостоверение;
7) игрушка, пищевой продукт, товар, молочный продукт;
8) квитанция, накладная, документ, чек;
9) автомобиль, поезд, транспортное средство, автобус;
10) двигатель, двигатель внутреннего сгорания, дизель, турбореактивный двигатель.

Контрольные вопросы

1 Что такое наследование, одиночное наследование, множественное наследование?

2 Какие объекты базового класса наследуются в производном, а какие нет?

3 На примере своей программы поясните механизм позднего связывания.

4 В каком случае C++ проигнорирует механизм виртуальных функций?

9 Лабораторная работа № 9. Дружественные функции и классы

Цель работы: получение навыков в разработке программ с использованием дружественных функций и классов.

Теоретические сведения

Дружественные функции и классы [1, с. 187–188].

Задания для самостоятельного выполнения

Каждый класс должен реализовывать метод show, который отображает на экране все свойства данного объекта.

Вызов дружественной функции должен сопровождаться выводом сообщения о том, какое свойство, какого класса было изменено и из какого класса был осуществлен вызов.

Вариант 1

1 Класс Машина (фирма, состояние, номер учета).

2 Класс Механик (имя, заработная плата).

Описание предметной области – механик может менять состояние машины на «исправен».

Вариант 2

1 Класс Машина (фирма, состояние, номер учета).

2 Класс Механик (имя, заработная плата).

3 Класс Инженер (имя).

Описание предметной области –инженер может менять номер учета машины и зарплату механика.

Вариант 3

1 Класс Пациент (имя, диагноз, рецепт, номер карты).

2 Класс Доктор (имя, специализация).

3 Класс Заведующий отделением (имя).

Описание предметной области – доктор может менять диагноз и рецепт, заведцющий отделением может менять номер карты.

Вариант 4

1 Класс Пациент (имя, диагноз, рецепт, номер карты).

2 Класс Доктор (имя, специализация).

Описание предметной области – доктор может менять диагноз и рецепт пациента.

Вариант 5

1 Класс Пациент (имя, диагноз, рецепт, номер карты).

2 Класс Зав. отделением (имя).

Описание предметной области – зав. отделением может менять номер карты пациента.

Вариант 6

1 Класс Зверь (имя, порода, клетка, корм).

2 Класс Ветеринар (имя, зарплата).

3 Класс Продавец (имя, зарплата).

Описание предметной области – ветеринар может менять корм, продавец может менять клетку.

Контрольные вопросы

1 Какие классы и функции называются дружественными?

2 Как осуществляется перегрузка операций?

3 Сколько аргументов требуется для определения перегруженной унарной (бинарной) операции?

4 Чем отличается перегруженная операция ++ при ее использовании в префиксной форме от использовании в постфиксной форме?

10 Лабораторная работа № 10. Шаблоны

Цель работы: получение навыков в разработке программ с использованием шаблонов классов.

Содержание отчета

1 Тема и цель работы.

2 Текст программы.

3 Результаты выполнения программы.

Теоретические сведения

Шаблоны [1, с. 211–221].

Задания для самостоятельного выполнения

1 Создать шаблон класса «стек». Написать программу, использующую этот шаблон класса для моделирования Т-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.

2 Создать шаблон класса «стек». Написать программу, использующую этот шаблон для отыскания прохода по лабиринту с использованием данного шаблона класса.

Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице. После отыскания прохода программа печатает найденный путь в виде координат квадратов.

3 Создать шаблон класса «стек». Написать программу, моделирующую процесс прибытия и отъезда машин с использованием этого шаблона класса.

Гаражная стоянка имеет одну стояночную полосу, причем въезд и выезд находятся в одном конце полосы. Если владелец автомашины приходит забрать свой автомобиль, который не является ближайшим к выходу, то все автомашины, загораживающие проезд, удаляются, машина данного владельца выводится со стоянки, а другие машины возвращаются на стоянку в исходном порядке.

Прибытие или отъезд автомашины задается командной строкой, которая содержит признак прибытия или отъезда и номер машины. Программа должна выводить сообщение при прибытии или выезде любой машины. При выезде автомашины со стоянки сообщение должно содержать число раз, которое машина удалялась со стоянки для обеспечения выезда других автомобилей.

4 Создать шаблон класса «однонаправленный линейный список». Написать программу, которая содержит динамическую информацию о наличии автобусов в автобусном парке с использованием данного шаблона класса.

Сведения о каждом автобусе содержат:

– номер автобуса;
– фамилию и инициалы водителя;
– номер маршрута.

Программа должна обеспечивать:

– начальное формирование данных о всех автобусах в парке в виде списка;
– при выезде каждого автобуса из парка вводится номер автобуса, и программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте;
– при въезде каждого автобуса в парк вводится номер автобуса, и программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке;
– по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте.

5 Создать шаблон класса «однонаправленный линейный список». Написать программу, которая содержит текущую информацию о заявках на авиабилеты, с использованием данного шаблона класса.

Каждая заявка содержит:

– пункт назначения;
– номер рейса;
– фамилию и инициалы пассажира;
– желаемую дату вылета.

Программа должна обеспечивать:

– хранение всех заявок в виде списка;
– добавление заявок в список;
– удаление заявок из списка;
– вывод заявок по заданному номеру рейса и дате вылета;
– вывод всех заявок.

6 Создать шаблон класса «бинарное дерево». Написать программу, которая содержит текущую информацию о книгах в библиотеке, с использованием данного шаблона класса.

Сведения о книгах содержат:

– номер УДК;
– фамилию и инициалы автора;
– название;
– год издания;
– количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

– начальное формирование данных о всех книгах в библиотеке в виде двоичного дерева;
– добавление данных о книгах, вновь поступающих в библиотеку;
– удаление данных о списываемых книгах;
– по запросу выдаются сведения о наличии книг в библиотеке, упорядоченные по годам издания.

7 Создать шаблон класса «бинарное дерево». Написать программу, которая содержит текущую информацию о заявках на авиабилеты, с использованием данного шаблона класса.

Каждая заявка содержит:

– пункт назначения;
– номер рейса;
– фамилию и инициалы пассажира;
– желаемую дату вылета.

Программа должна обеспечивать:

– хранение всех заявок в виде двоичного дерева;
– добавление и удаление заявок;
– удаление заявок из списка;
– вывод заявок по заданному номеру рейса и дате вылета с их последую-

щим удалением;

– вывод всех заявок.

8 Создать шаблон класса «бинарное дерево». Использовать его для сортировки целых чисел и строк, задаваемых с клавиатуры или из файла.

9 Создать шаблон класса «очередь». Написать программу, демонстрирующую работу с этим шаблоном для различных типов параметров шаблона. Программа должна содержать меню, позволяющее осуществить проверку всех методов шаблона.

10 Создать шаблон класса «очередь с приоритетами». При добавлении элемента в такую очередь его номер определяется его приоритетом. Написать программу, демонстрирующую работу с этим шаблоном для различных типов параметров шаблона. Программа должна содержать меню, позволяющее осуществить проверку всех методов шаблона.

11 Создать шаблон класса для работы с комплексными числами, обеспечивающий выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, использующую этот шаблон, задавая вещественную и мнимую части как числами типа double, так и целыми числами.

12 Создать шаблон класса одномерных массивов чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы массива, возможность задания произвольных границ индексов при создании объекта и выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножении и деления всех элементов массива на число, вывода на экран элемента массива по заданному индексу и всего массива.

Написать программу, использующую созданный шаблон для создания массивов различного типа.

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

Написать программу, использующую созданный шаблон для создания матриц различного типа.

14 Описать шаблон класса «множество», позволяющий выполнять основные операции – добавление и удаление элемента, пересечение, объединение и разность множеств.

Написать программу, демонстрирующую работу с этим шаблоном для различных типов данных.

15 Создать шаблон класса «однонаправленный кольцевой список». Написать программу, которая содержит текущую информацию о книгах в библиотеке, с использованием данного шаблона класса.

Сведения о книгах содержат:

– номер УДК;

– фамилию и инициалы автора;

– название;

– год издания;

– количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

– начальное формирование данных о всех книгах в библиотеке в виде однонаправленного кольцевого списка;

– при взятии каждой книги вводится номер УДК, и программа уменьшает значение количества книг на единицу или выдает сообщение о том, что требуемой книги в библиотеке нет или требуемая книга находится на руках;

– при возвращении каждой книги вводится номер УДК, и программа увеличивает значение количества книг на единицу;

– по запросу выдаются сведения о наличии книг в библиотеке.

Контрольные вопросы

1 Что такое шаблон?

2 Для чего может быть определен шаблон?

3 Расскажите о назначении шаблонов.

4 Приведите синтаксис объявления шаблона класса.

5 Может ли быть список параметров шаблона пустым?

6 Есть ли ошибка в определении шаблона: template<class T, class U>void fo o (T*, U)?

7 Может ли быть непараметризовано возвращаемое шаблоном функции значение? Приведите пример.

8 Можно ли перегружать шаблоны функции?

11 Лабораторная работа № 11. Организация работы с текстовыми файлами

Цель работы: получение навыков в разработке программ с использованием файлов. Получение навыков в разработке программ с использованием стандартного строкового класса string.

Содержание отчета

1 Тема и цель работы.

2 Текст программы.

3 Результаты выполнения программы.

Теоретические сведения

1 Стандартные потоки [1, с. 267–271].

2 Файловые потоки [1, с. 280–282].

3 Строковые потоки [1, с. 282–283].

4 Потоки и типы, определенные пользователем [1, с. 284–285].

Задания для самостоятельного выполнения

1 Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.

2 Написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие заданное с клавиатуры слово.

3 Написать программу, которая считывает текст из файла и выводит на экран только строки, содержащие двузначные числа.

4 Написать программу, которая считывает английский текст из файла и выводит на экран слова, начинающиеся с гласных букв.

5 Написать программу, которая считывает текст из файла и выводит его на экран, меняя местами каждые два соседних слова.

6 Написать программу, которая считывает текст из файла и выводит на экран только предложения, не содержащие запятых.

7 Написать программу, которая считывает текст из файла и определяет, сколько в нем слов, состоящих не более чем из четырех букв.

8 Написать программу, которая считывает текст из файла и выводит на экран только цитаты, т. е. предложения, заключенные в кавычки.

9 Написать программу, которая считывает текст из файла и выводит на экран только предложения, состоящие из заданного количества слов.

10 Написать программу, которая считывает английский текст из файла и выводит на экран слова текста, начинающиеся с гласных букв и оканчивающиеся гласными буквами.

11 Написать программу, которая считывает текст из файла и выводит на экран только строки, не содержащие двузначные числа.

12 Написать программу, которая считывает текст из файла и выводит на экран только предложения, начинающиеся с тире, перед которым могут следовать только пробельные символы.

13 Написать программу, которая считывает английский текст из файла и выводит его на экран, заменив каждую первую букву слов, начинающихся с гласной буквы, на прописную.

14 Написать программу, которая считывает текст из файла и выводит его на экран, заменив цифры от 0 до 9 на слова «ноль», «один», ..., «девять», начиная каждое предложение с новой строки.

15 Написать программу, которая считывает текст из файла, находит самое длинное слово и определяет, сколько раз оно встретилось в тексте.

Контрольные вопросы

1 Дайте определение потока.

2 Назовите три потоковых класса, предназначенных для файлового ввода-вывода.

3 Истинно ли утверждение о том, что некоторые потоки являются входными, а некоторые – выходными?

4 Напишите выражение, записывающее единичный символ в объект класса ofstream.

12 Лабораторная работа № 12. Организация работы с бинарными файлами

Цель работы: получение навыков в разработке программ с использованием бинарных файлов.

Содержание отчета

1 Тема и цель работы.
2 Текст программы.
3 Результаты выполнения программы.

Теоретические сведения

1 Стандартные потоки [1, с. 267–271].
2 Файловые потоки [1, с. 280–282].
3 Потоки и типы, определенные пользователем [1, с. 284–285].

Задания для самостоятельного выполнения

Для выполнения лабораторной работы использовать классы, разработанные в лабораторной работе № 10.

Для соответствующих классов перегрузить операции вставки в поток и извлечения из потока.

Массив объектов класса формировать путем чтения из файла.

По окончании работы массив объектов также сохранять в файл.

Контрольные вопросы

1 Объект какого класса следует использовать для того, чтобы выполнить чтение из файла с произвольной позиции?

2 Как определяется двоичный поток в стандарте языка С?

3 Как в языке С кодируется признак конца файла?

Как в языке С кодируется признак конца строки?

5 Как осуществляется запись бинарной информации в текстовый файл?

6 Как осуществляется чтение бинарной информации из текстового файла?

13 Лабораторная работа № 13. Исключения. Реализация механизма обработки исключений при разработке программ

Цель работы: получение навыков в разработке программ с использованием обработчиков исключительных ситуаций.

Содержание отчета

1 Тема и цель работы.
2 Текст программы.
3 Результаты выполнения программы.

Теоретические сведения

Обработка исключительных ситуаций [1, с. 222–230].

Задания для самостоятельного выполнения

Для выполнения лабораторной работы использовать классы, разработанные в лабораторной работе № 12.

Осуществлять контроль состояния потоков. В случае возникновения ошибок потоков генерировать и обрабатывать исключительные ситуации.

При динамическом выделении памяти предусмотреть обработку исключения, возникающего при нехватке памяти.

При вводе данных предусмотреть обработку исключения, возникающего при несоответствии типов данных.

Контрольные вопросы

1 Из-за чего в большинстве случаев возникает исключение:
– программиста, написавшего исходный код приложения;
– создателя класса, написавшего его методы;
– ошибки выполнения;
– сбоя в операционной системе?

2 Какие ключевые слова используются при работе с механизмом исключений в C++?

3. Напишите выражение, генерирующее исключение, используя класс BoundsError (тело класса пусто).

4 Истинно ли утверждение о том, что выражения, которые могут создать исключительную ситуацию, должны быть частью блока-ловушки?

5 Как передаются исключения:

– из блока-ловушки в блок повторных попыток;

– из выражения, создавшего исключительную ситуацию, в блок повторных попыток;

– из точки, где возникла ошибка, в блок-ловушку;

– из выражения, в котором возникла ошибка, в блок-ловушку?

6 Напишите спецификацию класса исключений, хранящего номер ошибки и ее название. Включите в класс конструктор.

7 Истинно ли утверждение о том, что выражение, генерирующее исключение, не должно быть расположено в блоке повторных попыток ?

8 Для каких ошибок обычно генерируется исключение:

– чрезмерное количество данных грозит переполнить массив;

– пользователь нажал Ctrl+C для закрытия программы;

– скачок напряжения в сети привел к перезагрузке системы;

– new не может зарезервировать необходимый объем памяти?

9 Куда может быть помещена дополнительная информация, передающаяся при генерации исключения:

– ключевое слово throw;

– функцию, вызвавшую ошибку;

– блок-ловушку;

– объект класса исключений?

10 Истинно ли утверждение о том, что программа может продолжить свое выполнение после возникновения исключительной ситуации?

14 Лабораторная работа № 14. Контейнерные классы

Цель работы: получение навыков в разработке программ с использованием контейнерных классов.

Содержание отчета

1 Тема и цель работы.

2 Текст программы.

3 Результаты выполнения программы.

Теоретические сведения

1 Контейнерные классы [1, с. 295–297].

2 Последовательные контейнеры [1, с. 297–299].

Задания для самостоятельного выполнения

В каждом упражнении требуется реализовать в том или ином виде динамические контейнеры. Для демонстрации работы с динамическим контейнером во всех заданиях надо написать главную функцию. В программе должны присутствовать различные способы создания объектов и массивов объектов. Программа должна демонстрировать использование всех функций и методов.

Во всех заданиях обязательно должны быть реализованы безаргументные и инициализирующие конструкторы, в том числе конструктор с двумя аргументами-итераторами, конструктор копирования, деструктор, операция присваивания, ввод-вывод. Подходящие операции реализуются как методы класса, а остальные – как внешние дружественные функции. Должна быть поддержана обработка исключений по нехватке памяти: все конструкторы обязаны иметь спецификацию исключений bad_alloc.

Размеры массива нужно задавать в конструкторе. Обязательно должны быть реализованы операция присваивания и соответствующие задаче операции с присваиванием; операция индексирования [] должна проверять индекс на допустимость и генерировать исключение в случае ошибки; операции с двумя массивами должны проверять совпадение размеров.

Варианты

1 Создать класс Decimal для работы с беззнаковыми целыми десятичными числами, используя массив элементов типа unsigned char, каждый элемент которого является десятичной цифрой. Младшая цифра имеет меньший индекс (единицы – в нулевом элементе массива). Реализовать арифметические операции, аналогичные имеющимся для целых в C++, и операции сравнения.

2 Создать класс Polinom для работы с многочленами. Коэффициенты должны быть представлены массивом, каждый элемент которого – коэффициент. Младшая степень имеет меньший индекс (нулевая степень – нулевой индекс). Реализовать арифметические операции, вычисление для заданного x, интегрирование, получение производной.

3 Создать класс Octal для работы с беззнаковыми целыми восьмеричными числами, используя массив элементов типа unsigned char, каждый элемент которого является шестнадцатеричной цифрой. Младшая цифра имеет меньший индекс. Реализовать все арифметические операции для целых в C++ и операции сравнения.

4 Создать класс Decimal для работы со знаковыми целыми десятичными числами, используя массив элементов типа unsigned char, каждый элемент которого является десятичной цифрой. Младшая цифра имеет меньший индекс (единицы – в нулевом элементе массива). Знак представить отдельным полем sign. Реализовать арифметические операции, встроенные для целых в C++, и операции сравнения.

5 Создать класс Fraction для работы с дробными десятичными числами.

Количество цифр в дробной части должно задаваться в отдельном поле и инициализироваться конструктором. Знак представить отдельным полем sign.

6 Создать класс Long для работы с беззнаковыми целыми двоичными числами, используя массив элементов типа unsigned char, каждый элемент которого является десятичной цифрой. Младшая цифра имеет меньший индекс (единицы – в нулевом элементе массива). Реализовать арифметические операции, встроенные для целых в C++, и операции сравнения.

7 Создать класс Hex для работы с беззнаковыми целыми шестнадцатеричными числами, используя массив элементов типа unsigned char, каждый элемент которого является шестнадцатеричной цифрой. Младшая цифра имеет меньший индекс. Реализовать арифметические операции, поддержанные для целых в C++, и операции сравнения.

8 Создать класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long – для рублей и типа unsigned char – для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Сумма должна быть представлена массивом, каждый элемент которого – десятичная цифра. Младший индекс соответствует младшей цифре денежной суммы. Младшие две цифры – копейки.

9 Создать класс Long для работы со знаковыми целыми двоичными числами, используя массив элементов типа unsigned char, каждый элемент которого является десятичной цифрой. Младшая цифра имеет меньший индекс (единицы – в нулевом элементе массива). Знак представить отдельным полем sign. Реализовать арифметические операции, поддержанные для целых в C++, и операции сравнения.

10 Создать класс Hex для работы со знаковыми целыми шестнадцатеричными числами, используя массив элементов типа unsigned char, каждый элемент которого является шестнадцатеричной цифрой. Младшая цифра имеет меньший индекс. Знак представить отдельным полем sign. Реализовать арифметические операции, поддержанные для целых в C++, и операции сравнения.

Контрольные вопросы

1 Что такое контейнер?
2 Что такое итератор?
3 Что такое функциональный объект?
4 Что такое адаптер контейнера?
5 Перечислите последовательные контейнеры STL.
6 Перечислите ассоциативные контейнеры STL.
7 Истинно ли утверждение о том, что одной из функций итераторов STL является связывание алгоритмов и контейнеров?
8 Истинно ли утверждение о том, что алгоритмы могут использоваться только с контейнерами STL?

15 Лабораторная работа № 15. Разработка собственного контейнерного класса по аналогии с классами STL

Цель работы: получение навыков в разработке программ с использованием стандартной библиотеки шаблонов STL.

Содержание отчета

1 Тема и цель работы.
2 Текст программы.
3 Результаты выполнения программы.

Теоретические сведения

1 Двусторонние очереди [1, с. 304–306].
2 Списки [1, с. 306–310].
3 Стеки [1, с. 310–311].
4 Очереди [1, с. 311–313].

Задания для самостоятельного выполнения

1 Написать программу для моделирования Т-образного сортировочного узла на железной дороге с использованием контейнерного класса stack из STL.

Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность ввода исходных данных с клавиатуры и из файла.

2 Написать программу, отыскивающую проход по лабиринту, с использованием контейнерного класса stack из STL.

Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Программа находит проход через лабиринт, двигаясь от заданного входа. После отыскания прохода программа выводит найденный путь в виде координат квадратов.

3 Написать программу, моделирующую управление каталогом в файловой системе.

Для каждого файла в каталоге содержатся следующие сведения: имя файла, дата создания, количество обращений к файлу.

Программа должна обеспечивать:

- начальное формирование каталога файлов;
- вывод каталога файлов;
- удаление файлов, дата создания которых раньше заданной;
- выборку файла с наибольшим количеством обращений.

Выбор моделируемой функции должен осуществляться с помощью меню. Для представления каталога использовать контейнерный класс list из STL.

4 Написать программу моделирования работы автобусного парка.

Сведения о каждом автобусе содержат номер автобуса, фамилию и инициалы водителя, номер маршрута.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

– начальное формирование данных о всех автобусах в парке в виде списка (ввод с клавиатуры или из файла);

– имитация выезда автобуса из парка: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте;

– имитация въезда автобуса в парк: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке;

– вывод сведений об автобусах, находящихся в парке, и об автобусах, находящихся на маршруте.

Для представления необходимых списков использовать контейнерный класс list.

5 Написать программу учета заявок на авиабилеты.

Каждая заявка содержит пункт назначения, номер рейса, фамилию и инициалы пассажира, желаемую дату вылета.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

– добавление заявок в список;

– удаление заявок;

– вывод заявок по заданному номеру рейса и дате вылета;

– вывод всех заявок.

Для хранения данных использовать контейнерный класс list.

6 Написать программу учета книг в библиотеке.

Сведения о книгах содержат фамилию и инициалы автора, название, год издания, количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

– добавление данных о книгах, вновь поступающих в библиотеку;

– удаление данных о списываемых книгах;

– выдача сведений о всех книгах, упорядоченных по фамилиям авторов;

– выдача сведений о всех книгах, упорядоченных по годам издания.

Хранение данных организовать с применением контейнерного класса multimap, в качестве ключа использовать «фамилию и инициалы автора».

7 Написать программу «Моя записная книжка».

Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения

или номеру телефона), добавления и удаления записей, сортировки по разным полям.

Хранение данных организовать с применением контейнерного класса map или multimap.

8 Написать программу учета заявок на обмен квартир и поиска вариантов обмена.

Каждая заявка содержит сведения о двух квартирах: требуемой (искомой) и имеющейся. Сведения о каждой квартире содержат: количество комнат, площадь, этаж, район.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

– ввод заявки на обмен;

– поиск в картотеке подходящего варианта: при совпадении требований и предложений по количеству комнат и этажности и различии по показателю «площадь» в пределах 10 % выводится соответствующая карточка и удаляется из списка, в противном случае поступившая заявка включается в картотеку;

– вывод всей картотеки.

Для хранения данных картотеки использовать контейнерный класс list.

9 Написать программу «Автоматизированная информационная система на железнодорожном вокзале».

Информационная система содержит сведения об отправлении поездов дальнего следования. Для каждого поезда указывается номер поезда, станция назначения, время отправления.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

– первоначальный ввод данных в информационную систему (с клавиатуры или из файла);

– вывод сведений по всем поездам;

– вывод сведений по поезду с запрошенным номером;

– вывод сведений по тем поездам, которые следуют до запрошенной станции назначения.

Хранение данных организовать с применением контейнерного класса vector.

10 Написать программу «Англо-русский и русско-английский словарь».

База данных словаря должна содержать синонимичные варианты перевода слов.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

– загрузка «базы данных» словаря (из файла);

– выбор режима работы (англо-русский, русско-английский);

– вывод вариантов перевода заданного английского слова;

– вывод вариантов перевода заданного русского слова.

Базу данных словаря реализовать в виде двух контейнеров типа map.

11 Написать программу, реализующую игру «Крестики-нолики» между двумя игроками: пользователем и компьютером (роботом). В программе использовать контейнерные классы STL.

12 Написать программу, решающую игру-головоломку «Игра в 15». Начальное размещение номеров – случайное. Предусмотреть два режима демонстрации решения: непрерывный (с некоторой задержкой визуализации) и пошаговый (по нажатию любой клавиши). В программе использовать контейнерные классы STL.

13 Составить программу формирования списка кандидатов, участвующих в выборах губернатора.

Каждая заявка от кандидата содержит фамилию и инициалы, дату рождения, место рождения, индекс популярности.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

– добавление заявки в список кандидатов. Для ввода индекса популярности (значение указано в скобках) предусмотреть выбор с помощью подменю одного из следующих вариантов:

– поддержан президентом (70);

– поддержан оппозиционной партией (15);

– оппозиционный кандидат, который снимет свою кандидатуру в пользу кандидата № 1 (10);

– прочие (5);

– удаление заявки по заявлению кандидата;

– формирование и вывод списка для голосования.

Хранение данных организовать с применением контейнерного класса priority_queue из STL. Для надлежащего функционирования очереди с приоритетами побеспокоиться о надлежащем определении операции < (меньше) в классе, описывающем заявку кандидата. Формирование и вывод списка для голосования реализовать посредством выборки заявок из очереди.

14 Составить программу моделирования работы автобусного парка.

Сведения о каждом автобусе содержат номер автобуса, фамилию и инициалы водителя, номер маршрута.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

– начальное формирование данных о всех автобусах в парке в виде списка (ввод с клавиатуры или из файла);

– имитация выезда автобуса из парка: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте;

– имитация въезда автобуса в парк: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке;

– вывод сведений об автобусах, находящихся в парке, и об автобусах, находящихся на маршруте, упорядоченных по номерам автобусов;

– вывод сведений об автобусах, находящихся в парке, и об автобусах, находящихся на маршруте, упорядоченных по номерам маршрутов.

Хранение всех необходимых списков организовать с применением контейнерного класса map, в качестве ключа использовать «номер автобуса».

15 Составить программу учета заявок на авиабилеты.

Каждая заявка содержит пункт назначения, номер рейса, фамилию и инициалы пассажира, желаемую дату вылета.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

– добавление заявок в список;
– удаление заявок;
– вывод заявок по заданному номеру рейса и дате вылета;
– вывод всех заявок, упорядоченных по пунктам назначения;
– вывод всех заявок, упорядоченных по датам вылета.

Хранение данных организовать с применением контейнерного класса multimap, в качестве ключа использовать «пункт назначения».

Контрольные вопросы

1 Какая сущность зачастую используется для изменения поведения алгоритма?

2 В каких случаях вектор является подходящим контейнером?

3 Для чего используется алгоритм unique()?

4 Напишите выражение, считывающее содержимое объекта ifile класса ifstream в массив buff.

16 Лабораторная работа № 16. Разработка приложений с использованием паттернов проектирования

Цель работы: получение навыков в разработке программ с использованием паттернов проектирования.

Теоретические сведения

1 Паттерн Builder [5, с. 102–107]
2 Паттерн Abstract Factory [5, с. 107–110]
3 Паттерн Factory Method [5, с. 111–115]

Задания для самостоятельного выполнения

1 Паттерн Builder. Имеется текст статьи в формате TXT. Статья состоит из заголовка, фамилий авторов, самого текста статьи и хеш-кода текста статьи. Написать приложение, позволяющее конвертировать документ в формате TXT в документ формата XML, необходимо также проверять корректность хеш-кода статьи.

2 Паттерн Abstract Factory. Разработать систему Кинопрокат. Пользователь может выбрать определённую киноленту, при заказе киноленты указывается язык звуковой дорожки, который совпадает с языком файла субтитров. Система должна поставлять фильм с требуемыми характеристиками, причём при смене языка звуковой дорожки должен меняться и язык файла субтитров и наоборот.

3 Паттерн Factory Method. Фигуры игры «тетрис». Реализовать процесс случайного выбора фигуры из конечного набора фигур. Предусмотреть появление супер-фигур с большим числом клеток, чем обычные.

4 Паттерн Builder. Разработать модель системы Автомобиль. Написать код приложения, который будет позволять порождать как серийные автомобили, так и автомобили по специальному заказу. Использовать шаблон.

5 Паттерн Builder. Разработать модель системы Музыкальный коллектив. Написать код приложения, позволяющий создавать певческие, танцевальные и смешанные коллективы.

6 Паттерн Builder. Разработать модель системы Комплексный обед. Написать код приложения, позволяющий создавать как стандартные комплексные обеды, так и обеды, в которые включены дополнительные блюда из меню.

7 Паттерн Abstract Factory. Написать код приложения, позволяющий универсально записывать данные о совершенном телефонном звонке в базу данных и xml-файл, а также считывать эту информацию. Использовать также шаблон DAO.

8 Паттерн Abstract Factory. Написать код приложения, позволяющий сохранять регистрационные данные пользователя в базе данных. Состав регистрационных данных у каждого пользователя может быть различен. Использовать также шаблон DAO.

Контрольные вопросы

1 Каково назначение паттерна Builder?
2 Какая проблема решается с помощью паттерна Builder?
3 В чём заключается решение, предлагаемое в паттерне Builder?
4 Для чего предназначен паттерн Abstract Factory?
5 Какое решение предлагается в паттерне Abstract Factory?
6 Какую проблему позволяет решить паттерн Factory Method?

Список литературы

1 **Павловская, Т. А.** C/C++. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. – Санкт-Петербург: Питер, 2016. – 461 с.

2 **Васильев, А. Н.** Программирование на C++ в примерах и задачах / А. Н. Васильев. – Москва: Эксмо, 2016. – 368 с.

3 **Шилдт, Г.** C++. Базовый курс / Г. Шилдт. – Москва: Вильямс, 2015. – 624 с.

4 **Полубенцева, М.** C/C++. Процедурное программирование / М. Полубенцева. – Санкт-Петербург: BHV, 2017. – 432 с.

5 **Тепляков, С.** Паттерны проектирования на платформе. NET / С. Тепляков. – Санкт-Петербург: Питер, 2015. – 320 с.