

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

*Методические рекомендации к лабораторным работам
для студентов специальности
1-40 05 01 «Информационные системы и технологии
(по направлениям)»
очной и заочной форм обучения*

Могилев 2021

УДК 621.01
ББК 36.4
О87

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой ««Программное обеспечение информационных технологий» «15» декабря 2020 г., протокол № 5

Составители: ст. преподаватель Ю. В. Вайнилович;
ст. преподаватель О. В. Сергиенко

Рецензент канд. техн. наук, доц. В. М. Ковальчук

Методические рекомендации к лабораторным работам предназначены для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)» очной и заочной форм обучения.

Учебно-методическое издание

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

| | |
|-------------------------|------------------|
| Ответственный за выпуск | В. В. Кутузов |
| Корректор | И. В. Голубцова |
| Компьютерная верстка | Н. П. Полевничая |

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, г. Могилев.

© Белорусско-Российский
университет, 2021

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. Дружественные методы | 7 |
| 3 Лабораторная работа № 3. Программирование алгоритмов с использованием механизма перегрузки операторов | 7 |
| 4 Лабораторная работа № 4. Программирование алгоритмов с использованием механизма простого наследования | 10 |
| 5 Лабораторная работа № 5. Реализация принципа полиморфизма. Виртуальные функции. Абстрактные классы | 11 |
| 6 Лабораторная работа № 6. Программирование алгоритмов с использованием механизма множественного наследования | 12 |
| 7 Лабораторная работа № 7. Шаблоны | 15 |
| 8 Лабораторная работа № 8. Практические приемы использования шаблонов классов. Реализация smart-указателей | 18 |
| 9 Лабораторная работа № 9. Практические приемы использования шаблонов классов. Реализация механизма транзакций | 19 |
| 10 Лабораторная работа № 10. Организация работы с потоками ввода- вывода. Состояние потока | 20 |
| 11 Лабораторная работа № 11. Организация работы с текстовыми файлами | 23 |
| 12 Лабораторная работа № 12. Организация работы с бинарными файлами..... | 25 |
| 13 Лабораторная работа № 13. Исключения. Реализация механизма обработки исключений при разработке программ | 26 |
| 14 Лабораторная работа № 14. Контейнерные классы. Разработка собственного контейнерного класса по аналогии с классами STL | 27 |
| 15 Лабораторная работа № 15. Реализация списков в виде контейнерного класса | 30 |
| 16 Лабораторная работа № 16. Разработка приложений с использованием паттернов проектирования | 34 |
| 17 Лабораторная работа № 17. Применение ООП в разработке прикладных программ..... | 36 |
| Список литературы..... | 39 |

Введение

При изучении дисциплины «Объектно-ориентированное программирование» студенты выполняют лабораторные работы, варианты которых приведены в данных 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 C++. 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 В соответствии с заданием составить описание класса. Реализовать метод печати данных класса и необходимые методы. Создать массив из объектов описанного класса. Вывести на экран значения полей элементов массива.

2 Реализовать метод, не указанный в задании. Разделить реализацию класса на несколько файлов.

3 Организовать ввод данных из файла.

Варианты заданий

1 Разработать приложение, позволяющее получить значения корней квадратного уравнения. Разработать класс Solution, который будет содержать три поля, предназначенные для хранения коэффициентов квадратного уравнения, и метод для получения решения квадратного уравнения.

2 Разработать приложение, позволяющее обрабатывать данные банковской карты. Разработать класс Kart, который будет содержать поля для хранения суммы и пинкода и номера карты, и методы для проверки пинкода и вывода суммы.

3 Разработать приложение, с помощью которого можно рассчитать распределение учебного времени по дисциплинам. Разработать класс Discipline, который инкапсулирует информацию об учебной дисциплине, название и количество часов, отводимых на каждый вид занятия (лекции, практические и самостоятельные работы), и методы для подсчета суммарного числа часов.

4 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 метод увеличения суммы на некое число.

5 Построить описание класса, содержащего информацию о почтовом адресе организации. Реализовать метод проверки на полноту введенного адреса у объектов этого класса и вывода соответствующего сообщения.

6 Составить описание класса для хранения данных о компьютерном питомце. Разработать класс Pet, который будет содержать поля, предназначенные для хранения данных о виде, имени, проценте здоровья, проценте сытости питомца. Реализовать методы кормления питомца (параметр – размер порции, в комментариях описать шкалу ценности порции) и вывода соответствующего сообщения.

7 Составить описание класса для объектов-векторов, задаваемых координатами концов в двумерном пространстве. Реализовать метод смещения вектора на числа dx и dy по осям OX и OY соответственно.

8 Составить описание класса для определения одномерных массивов целых чисел (векторов). Реализовать метод для замены всех чисел, удовлетворяющих некоторому условию (выбрать условие самостоятельно), на определенное значение.

9 Составить описание класса многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, вывод на экран описания многочлена (например, $2x^3 + 5x^2 - 7$).

10 Составить описание класса для обработки запланированных дел.

Разработать класс Task, который будет содержать поля, предназначенные для хранения даты и описания задания. Реализовать метод, который по введенной дате определяет, просрочено ли задание.

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. Дружественные методы

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 Изменить модификатор доступа к полям класса, описанного в предыдущей лабораторной работе. Разработать:
 - конструкторы класса, предусматривающие все варианты инициализации объектов. Продемонстрировать их вызов;
 - деструктор;
 - перегруженные методы печати закрытых членов класса;
 - функции доступа, необходимые по смыслу (не менее двух геттеров и не менее двух сеттеров).
- 2 Организовать проверку корректности вводимых в поля класса данных.
- 3 Добавить в деструктор сохранение в файл данных из полей объекта класса.

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 В соответствии с заданием составить описание класса. Реализовать конструктор. Выполнить перегрузку оператора вывода и требуемые возможности класса.

2 Реализовать перегрузку оператора, не указанного в задании. Желательно произвести перегрузку оператора того типа, который не был перегружен в первом задании. Например, унарного, если был перегружен бинарный. Или оператора сравнения, если перегружались только операторы арифметических действий. В отчете привести описание перегруженного оператора. Разделить реализацию класса на несколько файлов.

3 Реализовать перегрузку оператора, кроме вывода для класса из сквозного задания. В отчете привести описание перегруженного оператора.

Варианты заданий

1 Определить класс `Vector2` как вектор из четырех чисел типа `int`. Определить оператор `+` для сложения векторов и `*` для умножения вектора на число. Написать программу, демонстрирующую работу с этим классом.

2 Определить класс `Fraction` с полями из чисел типа `int`. Определить для класса бинарный оператор `+`, унарный `-` и `*` для умножения дроби на число. Написать программу, демонстрирующую работу с этим классом.

3 Определить класс `Power2` для работы с 2^n . Определить для класса оператор сравнения, префиксную и постфиксную формы оператора `++`, он должен увеличивать значение степени. Проверить корректность работы инкремента в выражениях с приоритетом операций.

Написать программу, демонстрирующую работу с этим классом.

4 Определить класс `Price` для работы с суммой, состоящей из рублей и копеек. Определить для класса бинарный оператор `+` для сложения с числами ти-

пов `int` и `double`.

Написать программу, демонстрирующую работу с этим классом.

5 Создать класс `Point` для работы с точками на плоскости. Определить для класса оператор сравнения расположения по оси `OX`. Определить для класса бинарный оператор `+` для сложения с числом типа `int`, смещающий точку на это число.

Написать программу, демонстрирующую работу с этим классом.

6 Создать класс `Time` для работы со временем в формате «час:минута:секунда». Определить для класса бинарный оператор `+` для сложения с числом типа `int` – количество секунд, учитывая, что можно задавать число больше 59. Определить для класса функцию преобразования времени в секунды `= int`.

Написать программу, демонстрирующую работу с этим классом.

7 Создать класс `Date` для работы со временем в формате «год:месяц:день». Определить для класса бинарный оператор `-` для нахождения разности между двумя датами в днях. Определить для класса унарный оператор `++` в префиксной форме для увеличения даты на один день, предусмотреть корректность результата.

Написать программу, демонстрирующую работу с этим классом.

8 Описать класс `Matrix`, реализующий тип данных «целочисленная матрица» и работу с ними. Класс должен реализовывать путем перегрузки операторов следующие операции над матрицами:

- сложение с другой матрицей и с числом число;
- операцию сравнения на равенство.

Написать программу, демонстрирующую работу с этим классом.

9 Описать класс `Matrix`, реализующий тип данных «целочисленная матрица» и работу с ними. Класс должен реализовывать путем перегрузки операторов следующие операции над матрицами:

- умножение матрицы на другое число;
- операцию сравнения на неравенство.

Написать программу, демонстрирующую работу с этим классом.

10 Разработать класс `Set` (множество). Внутренним представлением класса `Set` является целочисленный массив элементов множества. Класс должен обеспечивать следующие возможности:

- добавление элементов в множество с исключением дублирования элементов, путем перегрузки оператора `+`;
- выполнение операций сравнения множеств.

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 \geq 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 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 В соответствии с заданием из лабораторной работы № 1 выделить описание родительского и дочернего классов. Реализовать конструкторы.
- 2 Выполнить перегрузку оператора вывода и требуемые возможности классов.
- 3 Реализовать второй дочерний класс, используя сокрытие и модификацию родительских методов и данных.

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*.

5 Лабораторная работа № 5. 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 В соответствии с заданием из лабораторных работ № 1 и 4 выделить описание абстрактного родительского класса с виртуальными методами. Создать производный класс.

2 Создать второй производный класс.

3 Создать массив из объектов обоих производных классов и продемонстрировать работу виртуальных функций.

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 $p \rightarrow Fun()$; поставит на выполнение версию функции $Fun()$ из ... класса».

6 Лабораторная работа № 6. 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

1 Необходимо построить иерархию классов согласно схеме наследования, приведенной в варианте задания.

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

3 Функция $main$ должна иллюстрировать иерархию наследования.

Варианты заданий

Варианты заданий приведены на рисунке 1.

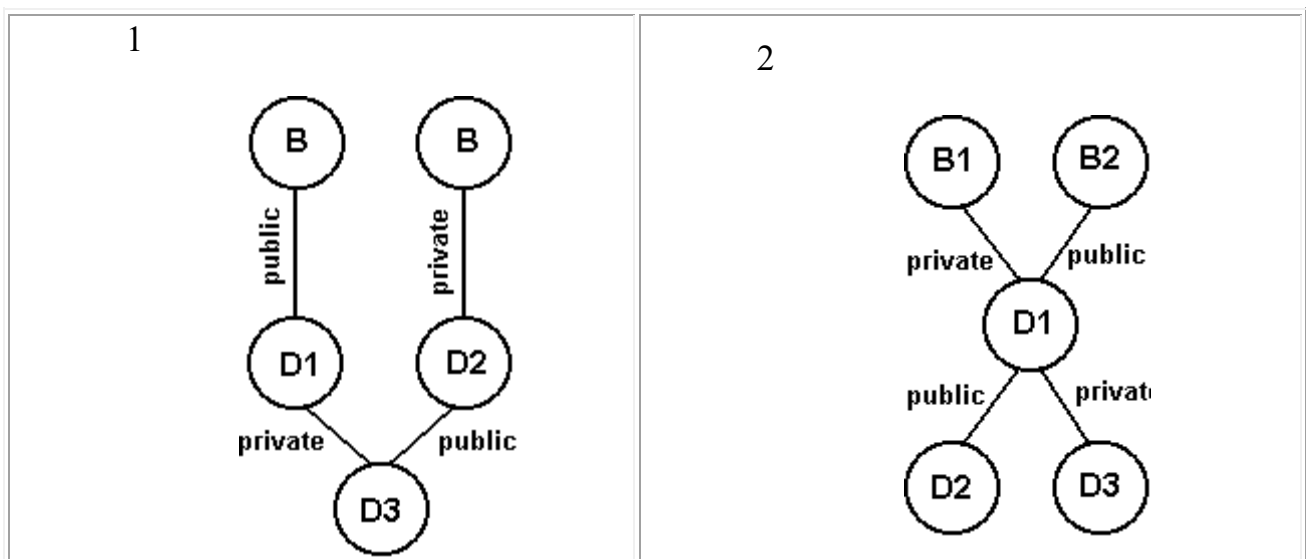
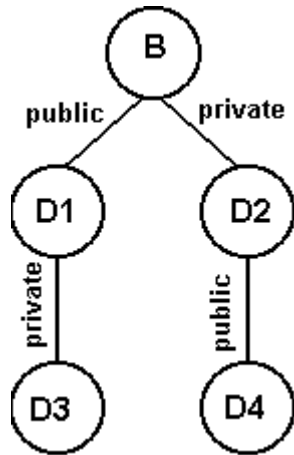
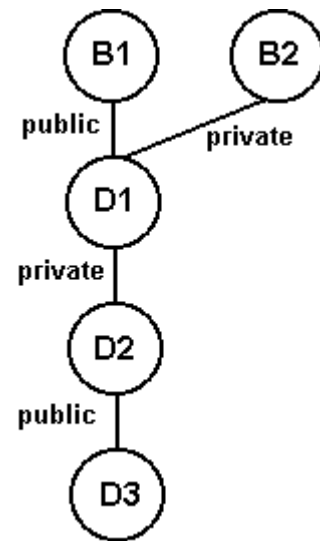


Рисунок 1 – Варианты заданий к лабораторной работе № 6

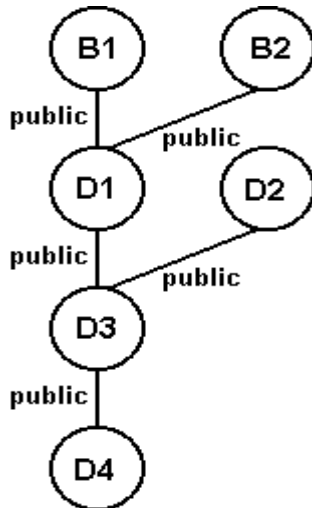
3



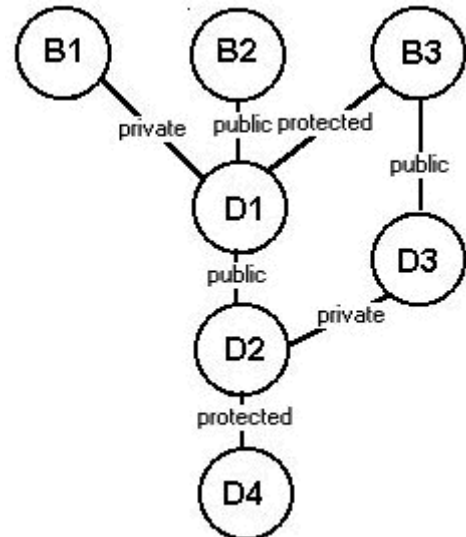
4



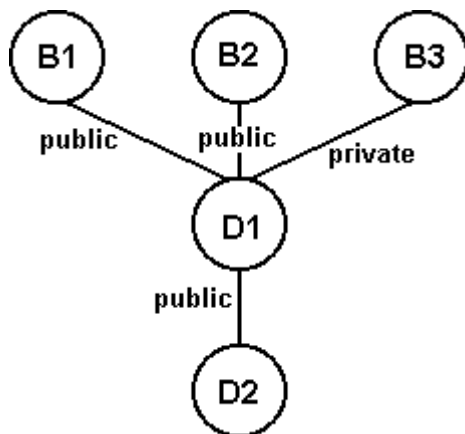
5



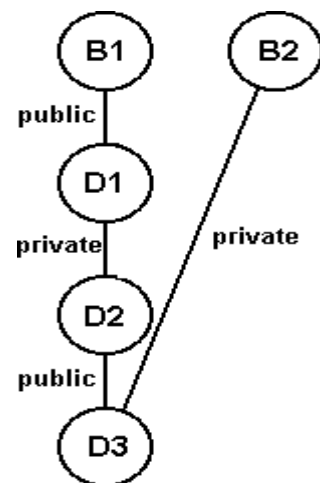
6

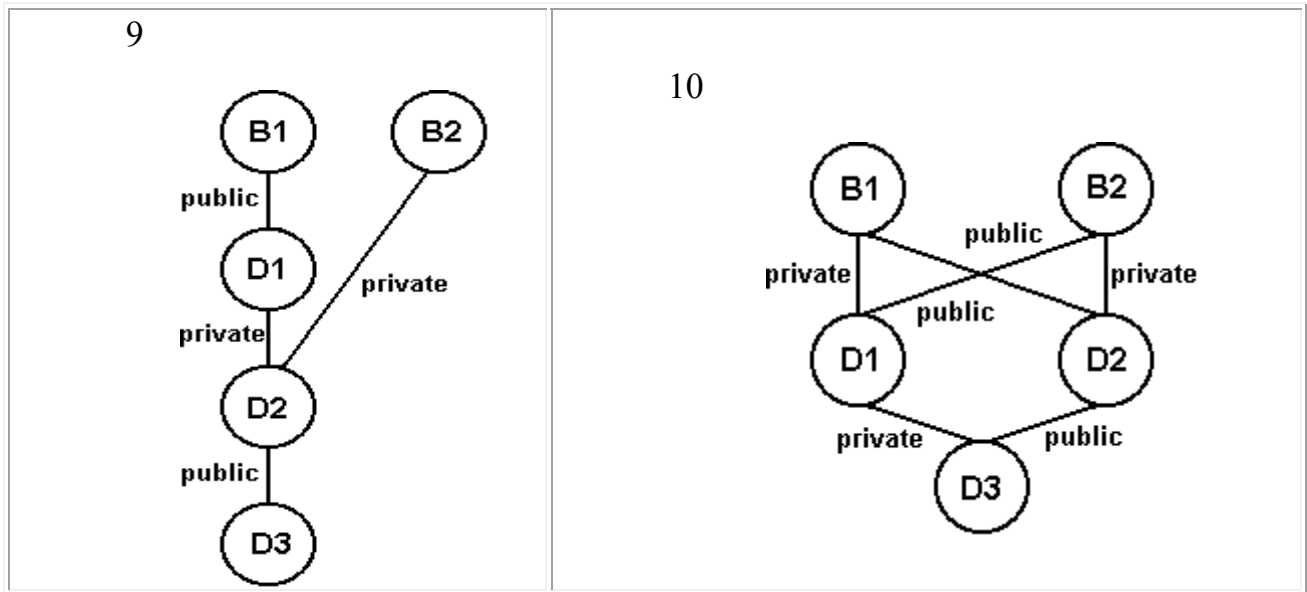


7



8





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?

7 Лабораторная работа № 7. Шаблоны

Цель работы

Получение навыков в разработке программ с использованием шаблонов классов.

Теоретические сведения

Шаблоны [1, с. 211–221].

Задания для самостоятельного выполнения

1 Создать шаблон класса «стек». Написать программу, использующую этот шаблон класса для моделирования T-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 использовани-

ем 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 последующим 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.

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 foo (T*, U)?`
- 7 Может ли быть непараметризовано возвращаемое шаблоном функции значение? Приведите пример.
- 8 Можно ли перегружать шаблоны функции?

8 Лабораторная работа № 8. Практические приемы использования шаблонов классов. Реализация smart-указателей

Цель работы

Получение навыков моделирования и реализации иерархии классов, а также использования статических данных и методов класса.

Теоретические сведения

Шаблоны [1, с. 211–221].

Задания для самостоятельного выполнения

Общая часть задания.

Для выполнения лабораторной работы использовать классы, разработанные в лабораторной работе № 2.

1 Создать несколько объектов класса с использованием smart-указателей класса `unique_ptr`. Применить к объектам методы класса. Уничтожить объекты путем уничтожения указателей.

2 Создать несколько объектов класса с использованием smart-указателей класса `shared_ptr`. Причем на один объект должно указывать несколько указателей. Число существующих указателей отслеживается при помощи счетчика ссылок. Уничтожить объекты.

3 Создать несколько объектов класса с использованием smart-указателей класса `weak_ptr`. Применить к объектам методы класса. Уничтожить объекты путем уничтожения указателей.

Контрольные вопросы

1 В каких случаях следует использовать `shared_ptr`, а в каких – `unique_ptr`?

2 Почему практически всегда следует использовать `make_shared` для создания объекта, которым будут впоследствии владеть `shared_ptr`?

3 Почему практически всегда следует использовать `make_unique` для создания объекта, которым будет владеть `unique_ptr`? Объясните.

4 Что такое `auto_ptr`?

5 Что такое циклическая ссылка?

9 Лабораторная работа № 9. Практические приемы использования шаблонов классов. Реализация механизма транзакций

Цель работы

Получение навыков в разработке программ с использованием шаблонов классов и механизма транзакций.

Теоретические сведения

Шаблоны [1, с. 211–221].

Задания для самостоятельного выполнения

К решению задачи из лабораторной работы № 8 добавить механизм транзакций – возможность закрепить или отменить выполненные изменения.

Для этого необходимо хранить состояние объекта на заданный момент, начало транзакции, и в момент принятия решения уничтожать предыдущее состояние (закрепление) или возвращаться к нему (отмена). Для обслуживания транзакций использовать smart-указатель. Основой его являются два указателя – один на текущий объект, а второй – на объект, представляющий его предыдущее состояние, и три функции – старт, закрепление, отмена.

Контрольные вопросы

- 1 Что называется транзакцией?
- 2 Назовите свойства транзакций.
- 3 Опишите стратегию работы с транзакциями.
- 4 В чем преимущество smart-указателей при работе с транзакциями?

10 Лабораторная работа № 10. Организация работы с потоками ввода-вывода. Состояние потока

Цель работы

Получение навыков в разработке программ с использованием файлов.
Получение навыков в разработке программ с использованием потоков ввода-вывода.

Содержание отчета

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

Теоретические сведения

- 1 Стандартные потоки [2, с. 266–283].
- 2 Файловые и строковые потоки. Строки класса `string` [2, с. 283–294].

Задания для самостоятельного выполнения

Вариант 1

1 Определить класс с именем `STUDENT`, содержащий следующие поля: фамилия и инициалы; номер группы; успеваемость (массив из пяти элементов).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа `STUDENT`.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти объектов типа `STUDENT`; записи должны быть упорядочены по возрастанию номера группы;
- вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0;
- если таких студентов нет, вывести соответствующее сообщение.

Вариант 2

1 Определить класс с именем `STUDENT`, содержащий следующие поля: фамилия и инициалы; номер группы; успеваемость (массив из пяти элементов).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа `STUDENT`.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти объектов типа `STUDENT`; записи должны быть упорядочены по возрастанию среднего балла;

- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5;
- если таких студентов нет, вывести соответствующее сообщение.

Вариант 3

1 Определить класс с именем STUDENT, содержащий следующие поля: фамилия и инициалы; номер группы; успеваемость (массив из пяти элементов).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа STUDENT.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти объектов типа STUDENT; записи должны быть упорядочены по алфавиту;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2;
- если таких студентов нет, вывести соответствующее сообщение.

Вариант 4

1 Определить класс с именем AEROFLOT, содержащий следующие поля: название пункта назначения рейса; номер рейса; тип самолета.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа AEROFLOT.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из семи объектов типа AEROFLOT; записи должны быть упорядочены по возрастанию номера рейса;
- вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры;
- если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Вариант 5

1 Определить класс с именем AEROFLOT, содержащий следующие поля: название пункта назначения рейса; номер рейса; тип самолета.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа AEROFLOT.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из семи объектов типа AEROFLOT; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
- вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры;
- если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Вариант 6

1 Определить класс с именем WORKER, содержащий следующие поля: фамилия и инициалы работника; название занимаемой должности; год поступления на работу.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа WORKER.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти объектов типа WORKER; записи должны быть размещены по алфавиту;
- вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;
- если таких работников нет, вывести на дисплей соответствующее сообщение.

Вариант 7

1 Определить класс с именем TRAIN, содержащий следующие поля: название пункта назначения; номер поезда; время отправления.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа TRAIN.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми объектов типа TRAIN; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
- вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени;
- если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 8

1 Определить класс с именем TRAIN, содержащий следующие поля: название пункта назначения; номер поезда; время отправления.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа TRAIN.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из шести объектов типа TRAIN; записи должны быть упорядочены по времени отправления поезда;
- вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры;
- если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 9

1 Определить класс с именем TRAIN, содержащий следующие поля: название пункта назначения; номер поезда; время отправления.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа TRAIN.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми объектов типа TRAIN; записи должны быть упорядочены по номерам поездов;
- вывод на экран информации о поезде, номер которого введен с клавиатуры;
- если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 10

1 Определить класс с именем MARSH, содержащий следующие поля: название начального пункта маршрута; название конечного пункта маршрута; номер маршрута.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа MARSH.

2 Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми объектов типа MARSH; записи должны быть упорядочены по номерам маршрутов;
- вывод на экран информации о маршруте, номер которого введен с клавиатуры;
- если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

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, то есть 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.

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 C++?
- 3 Как в языке C++ кодируется признак конца файла?
- 4 Как в языке C++ кодируется признак конца строки?
- 5 Как в языке C++ 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 Исключение в большинстве случаев возникает из-за:
 - а) программиста, написавшего исходный код приложения;
 - б) создателя класса, написавшего его методы;
 - в) ошибки выполнения;
 - г) сбоя в операционной системе.
- 2 При работе с механизмом исключений в C++ используются следующие ключевые слова: ...
- 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 Исключения передаются:
 - а) из блока-ловушки в блок повторных попыток;

б) из выражения, создавшего исключительную ситуацию, в блок повторных попыток;

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

г) 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 следующих ошибок обычно генерируется исключение:

а) чрезмерное количество данных грозит переполнить массив;

б) пользователь нажал Ctrl+C для закрытия программы;

в) скачок напряжения в сети привел к перезагрузке системы;

г) new не может зарезервировать необходимый объем памяти.

9 Дополнительная информация, передающаяся при генерации исключения, может быть помещена в:

а) ключевое слово throw;

б) функцию, вызвавшую ошибку;

в) блок-ловушку;

г) 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. Разработка собственного контейнерного класса по аналогии с классами STL

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, поддерживаемые 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, поддерживаемые 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, поддерживаемые 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. Реализация списков в виде контейнерного класса

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 T-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:

а) поддержан президентом (70);

б) поддержан оппозиционной партией (15);

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

г) 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 [6, с. 102–107].

2 Паттерн Abstract Factory [6, с. 107–110].

3 Паттерн Factory Method [6, 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 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 хэш-кода статьи.

8 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.

9 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.

10 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.

Контрольные вопросы

- 1 Каково назначение паттерна Строитель?
- 2 Какая проблема решается с помощью паттерна Строитель?
- 3 В чём заключается решение, предлагаемое в паттерне Строитель?
- 4 Для чего предназначен паттерн Абстрактная фабрика?
- 5 Какое решение предлагается в паттерне Абстрактная фабрика?
- 6 Какую проблему позволяет решить паттерн Фабричный метод?

17 Лабораторная работа № 17. Применение ООП в разработке прикладных программ

Цель работы

Получение навыков в разработке прикладных программ с использованием объектно-ориентированного программирования.

Теоретические сведения

Стандартные шаблоны [2, с. 294–332].

Задания для самостоятельного выполнения

Вариант 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.

Вариант 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.

Вариант 3

Написать программу, реализующую игру «Крестики-нолики» между двумя игроками: пользователь и компьютер (робот). В программе использовать контейнеры STL.

Вариант 4

Написать программу моделирования работы автобусного парка. Сведения о каждом автобусе содержат: номер автобуса, фамилию и инициалы водителя, номер маршрута. Следует обеспечить выбор с помощью меню и выполнение 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`.

Вариант 5

Написать программу учета заявок на авиабилеты. Каждая заявка содержит: пункт назначения, номер рейса, фамилию и инициалы пассажира, желаемую дату вылета. Следует обеспечить выбор с помощью меню и выполнение 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`.

Вариант 6

Написать программу учета книг в библиотеке. Сведения о книгах содержат: фамилию и инициалы автора, название, год издания, количество экземпляров данной книги в библиотеке. Следует обеспечить выбор с помощью меню и выполнение следующих функций:

- добавление данных о книгах, вновь поступающих в библиотеку;
- удаление данных о списываемых книгах;
- выдача сведений о всех книгах, упорядоченных по авторам;
- 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».

Вариант 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 помощью класса `map` или `multimap`.

Вариант 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. Следует обеспечить выбор с помощью меню и выполнение 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`.

Вариант 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. Следует обеспечить выбор с помощью меню и выполнение следующих функций:

- первоначальный ввод данных в систему (с клавиатуры или из файла);
- вывод сведений по всем поездам;
- вывод сведений по поезду с запрошенным номером;
- вывод сведений по поездам, следующим до запрошенной станции назначения.

Хранение данных организовать с применением контейнерного класса `vector`.

Вариант 10

Написать программу «Англо-русский и русско-английский словарь». «База данных» словаря должна содержать синонимичные варианты перевода слов. Следует обеспечить выбор с помощью меню и выполнение следующих функций:

- загрузка «базы данных» словаря (из файла);
 - выбор режима работы: англо-русский или русско-английский;
 - вывод вариантов перевода заданного английского слова;
 - вывод вариантов перевода заданного русского слова.
- Базу данных словаря реализовать в виде двух контейнеров типа map.

Контрольные вопросы

- 1 Какие изученные технологии ООП были использованы в работе?
- 2 Какие стандартные шаблоны были использованы?
- 3 Какие способы поддержки корректности работы программы были применены?

Список литературы

- 1 **Павловская, Т. А.** С/С++. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. – Санкт-Петербург: Питер, 2016. – 461 с.
- 2 **Павловская, Т. А.** С/С++. Структурное и объектно-ориентированное программирование: практикум / Т. А. Павловская, Ю. А. Щупак. – Санкт-Петербург: Питер, 2011. – 352 с.: ил.
- 3 **Васильев, А. Н.** Программирование на С++ в примерах и задачах / А. Н. Васильев. – Москва : Эксмо, 2016. – 368 с.
- 4 **Шилдт, Г.** С++. Базовый курс / Г. Шилдт. – Москва: Вильямс, 2015. – 624 с.
- 5 **Полубенцева, М.** С/С++. Процедурное программирование / М. Полубенцева. – Санкт-Петербург: ВHV, 2017. – 432 с.
- 6 **Тепляков, С.** Паттерны проектирования на платформе .NET / С. Тепляков. – Санкт-Петербург: Питер, 2015. – 320 с.