

МИНОБРНАУКИ РОССИИ

Орский гуманитарно-технологический институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования «Оренбургский государственный университет»
(Орский гуманитарно-технологический институт (филиал) ОГУ)

Кафедра программного обеспечения

Методические указания по выполнению и защите лабораторных и практических работ
по дисциплине «Б1.Д.Б.21 Объектно-ориентированное программирование»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.03 Прикладная информатика
(код и наименование направления подготовки)

Прикладная информатика в экономике
(наименование направленности (профиля) образовательной программы)

Тип образовательной программы

Программа бакалавриата

Квалификация

Бакалавр

Форма обучения

Очная

Год начала реализации программы (набора)

2019

г. Орск 2018

Методические указания предназначены для обучающихся очной формы обучения направления подготовки 09.03.03 Прикладная информатика профилю Прикладная информатика в экономике по дисциплине «Б1.Д.Б.21 Объектно-ориентированное программирование»

Составитель _____



О.В. Подсобляева

Методические указания рассмотрены и одобрены на заседании кафедры программного обеспечения, протокол № 1 от «01» сентября 2018 г.

Заведующий кафедрой _____



Е.Е. Сурина

Согласовано:

Председатель методической комиссии по направлению подготовки 09.03.03 Прикладная информатика



Е.Е.Сурина

«12» сентября 2018 г.

© Подсобляева О.В., 2018
© Орский гуманитарно-технологический институт (филиал) ОГУ, 2018

Пояснительная записка

В результате изучения дисциплины «Б1.Д.Б.21 Объектно-ориентированное программирование» у обучающихся должны быть сформированы знания, умения и навыки:

- изучить основные принципы объектно-ориентированного программирования; изучить реализацию этих принципов на языках C++ и Java;

- научиться писать программы на языках C++ и Java;

- научиться проектировать и разрабатывать объектно-ориентированные программы.

Одной из наиболее эффективных форм закрепления теоретических знаний и выработки навыков самостоятельной работы являются лабораторные занятия.

Целью проведения лабораторных и практических занятий является:

- закрепление знаний студентов по основам проектной деятельности,

- формирование у студентов навыков использования современных технических средств и технологий для решения проектных и исследовательских задач.

Тематический план

Таблица 1 – Тематический план выполнения лабораторных и практических работ по дисциплине «Б1.Д.Б.21 Объектно-ориентированное программирование» для обучающихся направления подготовки 09.03.03 Прикладная информатика профиль подготовки Прикладная информатика в экономике

Лабораторные работы

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
1	1-3	Язык C++. Объекты и классы.	4
2	4	Язык C++. Конструкторы и деструкторы.	4
3	5	Язык C++. Массивы объектов, указатели и ссылки на объекты.	4
4	6	Язык C++. Перегрузка операций.	4
5	7	Язык C++. Наследование.	4
6	8	Язык C++. Виртуальные функции.	4
7	9	Язык C++. Потоки и файлы.	4
8	10	Язык C++. Многофайловые программы.	4
		Итого:	32

Практические работы

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
1	1-3	Язык C++. Объекты и классы.	6
2	4	Язык C++. Конструкторы и деструкторы.	6
3	5	Язык C++. Массивы объектов, указатели и ссылки на объекты.	6
4	6	Язык C++. Перегрузка операций.	6
5	7	Язык C++. Наследование.	6
6	8	Язык C++. Виртуальные функции.	6
7	9	Язык C++. Потоки и файлы.	6
8	10	Язык C++. Многофайловые программы.	6
		Итого:	48

Методические указания по выполнению и оформлению лабораторных и практических работ

Лабораторные и практические работы по дисциплине «Объектно-ориентированное программирование» предполагают решение задач по темам, представленным в тематическом плане.

В практической и лабораторной работе должны быть выполнены все предусмотренные задания. В работе должна просматриваться логическая последовательность и взаимная увязка основных частей работы.

Рекомендуемая структура работ:

1) цель работы;

2) задание в соответствии с выбранным вариантом;

3) теоретическая часть, включающая краткое изложение теоретических положений по теме практической работы, формулы для решения задания;

4) практическая часть, включающая решение задания по теме практической работы. Дополнительно для наглядности расчетный материал может быть представлен в виде таблиц, графиков;

5) выводы по работе;

6) список использованной литературы.

Работы могут быть оформлены:

- машинописным текстом на листах формата А4.

Титульный лист оформляется на основе СТО 02069024. 101 – 2014 «РАБОТЫ СТУДЕНЧЕСКИЕ. Общие требования и правила оформления».

Работа защищается устно и принимается к зачету, если нет замечаний по ее выполнению и оформлению. При отсутствии зачтенных лабораторных работ студент не допускается к зачету по дисциплине «Б1.Д.Б.21 Объектно-ориентированное программирование».

Лабораторная работа №1 Практическая работа №1 Язык C++. Объекты и классы.

Разработать классы для описанных ниже объектов. Включить в класс методы `set (...)`, `get (...)`, `show (...)`. Определить другие методы.

1. **Student**: Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс. Создать массив объектов. Вывести:

а) список студентов заданного факультета; б) списки студентов для каждого факультета и курса;

в) список студентов, родившихся после заданного года.

2. **Abiturient**: Фамилия, Имя, Отчество, Адрес, Оценки. Создать массив объектов. Вывести:

а) список абитуриентов, имеющих неудовлетворительные оценки; б) список абитуриентов, сумма баллов у которых не меньше заданной;

в) выбрать N абитуриентов, имеющих самую высокую сумму баллов, и список абитуриентов, имеющих полупроходной балл.

3. **Aeroflot**: Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели. Создать массив объектов. Вывести:

а) список рейсов для заданного пункта назначения; б) список рейсов для заданного дня недели;

в) список рейсов для заданного дня недели, время вылета для которых больше заданного.

4. **Book**: Автор, Название, Издательство, Год, Количество страниц. Создать массив объектов. Вывести:

а) список книг заданного автора; б) список книг, выпущенных заданным издательством;

в) список книг, выпущенных после заданного года.

5. **Worker**: Фамилия и инициалы, Должность, Год поступления на работу, Зарплата. Создать массив объектов. Вывести:
- а) список работников, стаж работы которых на данном предприятии превышает заданное число лет;
 - б) список работников, зарплата которых больше заданной;
 - в) список работников, занимающих заданную должность.
6. **Train**: Пункт назначения, Номер поезда, Время отправления, Число общих мест, Купейных, Плацкартных. Создать массив объектов. Вывести:
- а) список поездов, следующих до заданного пункта назначения;
 - б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;
 - в) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.
7. **Product**: Наименование, Производитель, Цена, Срок хранения, Количество. Создать массив объектов. Вывести:
- а) список товаров для заданного наименования;
 - б) список товаров для заданного наименования, цена которых не превышает указанной;
 - в) список товаров, срок хранения которых больше заданного.
8. **Patient**: Фамилия, Имя, Отчество, Адрес, Номер медицинской карты, Диагноз. Создать массив объектов. Вывести:
- а) список пациентов, имеющих данный диагноз;
 - б) список пациентов, номер медицинской карты которых находится в заданном интервале.
9. **Bus**: Фамилия и инициалы водителя, Номер автобуса, Номер маршрута, Марка, Год начала эксплуатации, Пробег. Создать массив объектов. Вывести:
- а) список автобусов для заданного номера маршрута;
 - б) список автобусов, которые эксплуатируются больше 10 лет;
 - в) список автобусов, пробег у которых больше 10 000 км.
10. **Customer**: Фамилия, Имя, Отчество, Адрес, Телефон, Номер кредитной карточки, Номер банковского счета. Создать массив объектов. Вывести:
- а) список покупателей в алфавитном порядке;
 - б) список покупателей, номер кредитной карточки которых находится в заданном интервале.
11. **File**: Имя файла, Размер, Дата создания, Количество обращений. Создать массив объектов. Вывести:
- а) список файлов, упорядоченный в алфавитном порядке;
 - б) список файлов, размер которых превышает заданный;
 - в) список файлов, число обращений к которым превышает заданное.
12. **Word**: Слово, Номера страниц, на которых слово встречается (от 1 до 10), Число страниц. Создать массив объектов. Вывести:
- а) слова, которые встречаются более чем на N страницах;
 - б) слова в алфавитном порядке;
 - в) для заданного слова номера страниц, на которых оно встречается.
13. **House**: Адрес, Этаж, Количество комнат, Площадь. Создать массив объектов. Вывести:
- а) список квартир, имеющих заданное число комнат;
 - б) список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в определенном промежутке;
 - в) список квартир, имеющих площадь, превосходящую заданную.
14. **Phone**: Фамилия, Имя, Отчество, Адрес, Номер, Время внутригородских разговоров, Время междугородних разговоров. Создать массив объектов. Вывести:
- а) сведения об абонентах, время внутригородских разговоров которых превышает заданное;
 - б) сведения об абонентах, воспользовавшихся междугородней связью;
 - в) сведения об абонентах, выведенные в алфавитном порядке.
15. **Person**: Фамилия, Имя, Отчество, Адрес, Пол, Образование, Год рождения. Создать массив объектов. Вывести:
- а) список граждан, возраст которых превышает заданный;
 - б) список граждан с высшим образованием;
 - в) список граждан мужского пола.

Лабораторная работа №2
Практическая работа №2
Язык C++. Конструкторы и деструкторы

Разработать перечисленные ниже классы. При разработке каждого класса возможны два варианта решения: а) данные-члены класса представляют собой переменные и массивы фиксированной размерности; б) память для данных-членов класса выделяется динамически.

1. «**Комплексное число**» – **Complex**. Класс должен содержать несколько конструкторов и операции для сложения, вычитания, умножения, деления, присваивания. Создать два вектора размерности n из комплексных координат. Передать их в функцию, которая выполняет сложение комплексных векторов.

2. Определить класс «**Дробь**» – **Fraction** в виде пары m, n . Класс должен содержать несколько конструкторов. Реализовать методы для сложения, вычитания, умножения и деления дробей. Перегрузить операции сложения, вычитания, умножения, деления, присваивания и операции отношения. Создать массив объектов и передать его в функцию, которая изменяет каждый элемент массива с четным индексом путем добавления следующего за ним элемента массива.

3. Разработать класс «**Вектор**» – **Vector** размерности n . Определить несколько конструкторов, в том числе конструктор копирования. Реализовать методы для вычисления модуля вектора, скалярного произведения, сложения, вычитания, умножения на константу. Перегрузить операции сложения, вычитания, умножения, инкремента, декремента, индексирования, присваивания для данного класса. Создать массив объектов. Написать функцию, которая для заданной пары векторов будет определять, являются ли они коллинеарными или ортогональными.

4. Определить класс «**Квадратная матрица**» – **Matrix**. Класс должен содержать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для сложения, вычитания, умножения матриц; вычисления нормы матрицы. Перегрузить операции сложения, вычитания, умножения и присваивания для данного класса. Создать массив объектов класса **Matrix** и передать его в функцию, которая изменяет i -ю матрицу путем возведения ее в квадрат. В головной программе вывести результат.

5. Разработать класс «**Многочлен**» – **Polynom** степени n . Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для вычисления значения полинома; сложения, вычитания и умножения полиномов. Перегрузить операции сложения, вычитания, умножения, инкремента, декремента, индексирования, присваивания. Создать массив объектов класса. Передать его в функцию, вычисляющую сумму полиномов массива и возвращающую полином-результат, который выводится на экран в головной программе.

6. Определить класс «**Стек**» – **Stack**. Элементы стека хранятся в массиве. Если массив имеет фиксированную размерность, то предусмотреть контроль выхода за пределы массива. Если память выделяется динамически и ее не хватает, то увеличить размер выделенной памяти. Включение элементов в стек и их извлечение реализовать как

виде методов, так и с помощью перегруженных операций. Создать массив объектов класса **Stack**. Передавать объекты в функцию, которая удаляет из стека первый (сверху), третий, пятый и т. д. элементы.

7. Построить классы для описания плоских фигур: круг, квадрат, прямоугольник. Включить методы для изменения объектов, перемещения на плоскости, вращения. Перегрузить операции, реализующие те же действия. Выполнить тестирование класса, создав массив объектов.

8. Определить класс «**Строка**» – **String** длины n . Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для выполнения конкатенации строк, извлечения символа из заданной позиции, сравнения строк. Перегрузить операции сложения, индексирования, отношения, добавления, присваивания.

ния для данного класса. Создать массив объектов и передать его в функцию, которая выполняет сортировку строк.

9. Разработать класс «Множество (целых чисел, символов, строк

ит. д.)» – **Set** мощности n . Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для определения принадлежности заданного элемента множеству, пересечения, объединения, разности двух множеств. Перегрузить операции сложения, вычитания, умножения (пересечения), индексирования, присваивания. Создать массив объектов и передавать пары объектов в функцию, которая строит множество, состоящее из элементов, входящих только в одно из заданных множеств, т. е. $A \setminus B$, и возвращает его в главную программу.

10. Разработать класс для массива строк. Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для поэлементной конкатенации двух массивов, упорядочения строк в лексикографическом порядке, слияния двух массивов с удалением повторяющихся строк, а также для вывода на экран всего массива и заданной строки. Перегрузить операции сложения, умножения, индексирования, присваивания для данного класса. Создать массив объектов и передавать объекты в функцию, которая выполняет слияние объектов и для полученного объекта-результата производит лексикографическое упорядочение строк.

11. Составить описание класса, обеспечивающего представление

матрицы заданного размера n и m и любого минора в ней. Память для матрицы выделять динамически. Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для отображения на экране как матрицы в целом, так и заданного минора, а также для изменения минора: сложения, вычитания, умножения миноров. Перегрузить операции сложения, вычитания, умножения и присваивания для данного класса. Создать массив объектов данного класса и передать его в функцию, которая изменяет для i -й матрицы

ее минор путем умножения на константу.

12. Построить класс «**Булев вектор**» – **BoolVector** размерности n . Определить несколько конструкторов, в том числе конструктор копирования. Реализовать методы для выполнения поразрядных конъюнкции, дизъюнкции и отрицания векторов, а также подсчета числа единиц и нулей в векторе. Реализовать те же действия над векторами с помощью перегруженных операций. Перегрузить операции отношения и присваивания для данного класса. Создать массив объектов. Передавать объекты в функцию, которая будет их изменять по формуле

13. Реализовать класс «**Троичный вектор**» – **Tvector** размерности n . Компоненты вектора принимают значения из множества $\{0, 1, X\}$.

Два троичных вектора $tk=(t1k, \dots, tnk)$ и $tl=(t1l, \dots, tnl)$ называются ортогональными, если существует такое i , что $tik, til \in \{0, 1\}$ и $tik \neq til$. Операция пересечения не ортогональных векторов выполняется покомпонентно по следующим правилам: $1 \cdot 1 = 1$, $X \cdot X = 1$, $0 \cdot 0 = 0$, $X \cdot 0 = 0$, $X \cdot X = X$.

Реализовать методы для проверки векторов на ортогональность, для пересечения не ортогональных векторов, сравнения векторов, подсчета числа компонент, равных X . Осуществить те же действия над векторами с помощью перегруженных операций. Перегрузить операцию присваивания для данного класса. Выполнить тестирование класса, создав массив объектов.

14. Определить класс «**Булева матрица**» – **BoolMatrix** размерности n и m . Класс должен содержать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для логического сложения (дизъюнкции), умножения и инверсии матриц. Реализовать методы для подсчета числа единиц в матрице и лексикографического упорядочения строк. Перегрузить операции для логического сложения, умножения и инверсии матриц, а также операцию присваивания. Создать массив объектов класса **BoolMatrix**.

Лабораторная работа №3
Практическая работа №3
Язык C++. Массивы объектов, указатели и ссылки на объекты.

При решении задач необходимо описать класс, который используется для представления элементов динамической структуры данных. Затем разрабатывается класс для работы с используемой динамической структурой данных, которая при тестировании класса может быть построена путем ввода данных: а) с клавиатуры; б) из файла. Возможны два варианта решения:

а) динамическая структура данных постоянно хранится в памяти; б) динамическая структура данных хранится в файле.

1. Создать класс для работы со стеком. Элемент стека – действительное число. Применить класс для вывода возрастающих серий последовательности действительных чисел: а) в обратном порядке; б) в том же порядке (серия – упорядоченная последовательность максимальной длины).

2. Построить класс для работы со стеком. Элемент стека – целое число. Ввести две неубывающие последовательности чисел в два стека. Использовать третий стек для слияния двух последовательностей в одну неубывающую.

3. Создать класс для работы со стеком. Элемент стека – символ. Сформировать два стека, содержащие последовательности символов. Подсчитать общее число элементов в стеках, предусмотреть восстановление их исходного расположения.

4. Создать класс для работы со стеком. Элемент стека – символ. Использовать стек для проверки правильности расстановки скобок трех типов (круглых, квадратных и фигурных) в выражении.

5. Построить класс для работы с односвязным списком. Элемент списка – действительное число. Сформировать список, содержащий неубывающую последовательность чисел, и преобразовать его так, чтобы последовательность была невозрастающей. Для этого необходимо совершить переворот списка, т. е. такую переустановку указателей в списке, при которой элементы его следуют друг за другом в обратном порядке.

6. Построить класс для работы с односвязным списком. Элементы списка – целые числа. Сформировать список, упорядочить элементы списка по возрастанию, используя сортировку: а) методом выбора; б) методом пузырька; в) методом вставки.

7. Построить класс для работы с односвязным списком. Элементы списка – действительные числа. Создать два упорядоченных по невозрастанию списка, слить их в один (также упорядоченный по невозрастанию), построив новый список.

8. Построить класс для работы с односвязным списком. Элементы списка – слова. Создать список, содержащий некоторую последовательность слов. Заменить в списке каждое вхождение заданного слова другим (также заданным).

9. Построить класс для работы с односвязным списком. Создать два списка: List1 и List2. Проверить, содержатся ли элементы списка List1 в списке List2 в указанном списке List1 порядке.

10. Построить класс для работы с односвязным списком. Элементы списка – целые числа. Создать список List1. Построить список List2, содержащий порядковые номера максимальных элементов списка List1.

11. Построить класс для работы с двусвязным списком. Элементы списка – действительные числа. Создать список List1, содержащий последовательность x_1, x_2, \dots, x_n . Построить список List2, содержащий последовательность $x_1, x_n, x_2, x_{n-1}, \dots, x_n, x_1$.

12. Создать класс для работы с бинарным деревом, узлы которого содержат целые числа. Построить дерево, затем копию дерева. Подсчитать число листьев в нем (листьями называются узлы, не содержащие поддеревьев).

13. Построить класс для работы с бинарным деревом, узлы которого содержат действительные числа. Создать дерево. Определить высоту дерева (максимальное число узлов,

принадлежащих пути от корня дерева до любого из его листьев). Подсчитать число элементов, равных максимальному.

14. Построить класс для работы с бинарным деревом, узлы которого содержат действительные числа. Создать дерево для заданной последовательности чисел. Используя его, упорядочить последовательность по возрастанию, убыванию.

15. Построить класс для работы со списком. Элемент списка содержит информацию о заявке на авиабилет: 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, вывод заявок по заданному номеру рейса и дате вылета, вывод всех заявок.

16. Построить класс для работы с бинарным деревом, узел которого содержит информацию о заявках на авиабилеты (в последовательности, используемой для упорядочения заявок): желаемую дату вылета, номер рейса, фамилию и инициалы пассажира, пункт назначения.

Программа должна обеспечивать: хранение всех заявок в виде бинарного дерева, добавление заявок, удаление заявок, вывод заявок по заданному номеру рейса и дате вылета, вывод всех заявок.

17. Построить класс для работы со списком, который содержит динамическую информацию о наличии автобусов в парке: номер автобуса, фамилию и инициалы водителя, номер маршрута, признак местонахождения автобуса – на маршруте или в парке.

Программа должна обеспечивать: начальное формирование списка, введение номера автобуса при выезде и установление программой значения признака «автобус на маршруте». Аналогичным образом изменяется информация об автобусе при его возвращении в парк. По запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте.

18. Решить предыдущую задачу, используя не список, а бинарное дерево.

19. Построить класс для работы с бинарным деревом, содержащим англо-русский словарь.

20. Построить класс для работы со списком, содержащим информацию о поездах дальнего следования. Элемент списка содержит следующую информацию о поезде: номер поезда, станция назначения, время отправления.

Составить программу, которая обеспечивает первоначальное формирование списка; производит вывод списка, вводит номер поезда и выводит информацию о нем, вводит название станции назначения и выводит данные о всех поездах, следующих до этой станции.

Лабораторная работа №4 **Практическая работа №4** **Язык C++. Перегрузка операций.**

Для разработки шаблонов классов можно использовать результаты выполнения лабораторных работ № 2 и № 3. При тестировании созданных шаблонов классов необходимо создавать объекты с различными допустимыми значениями параметров шаблона (например, компоненты вектора могут быть целыми, действительными или комплексными числами).

1. Создать шаблон класса для работы со стеком. Применить его для решения задач № 1 – 4 (лаб. работа № 3).

2. Создать шаблон класса для работы с одномерным массивом. Выполнить тестирование путем создания и обработки массивов, содержащих элементы различных типов (например, сортировка элементов массивов различными методами).

3. Создать шаблон класса **Vector** размерности n (см. задачу № 3, лаб. работа № 2).

4. Создать шаблон класса «Квадратная матрица» – **Matrix** размерности $n \times n$ (см. задачу № 4, лаб. работа № 2).

5. Создать шаблон класса **Polynom** степени n (см. задачу № 5, лаб. работа № 2) или создать шаблон класса для работы с односвязным списком. Применить его для решения задачи № 5 (лаб. работа № 3).

6. Создать шаблон класса для работы с односвязным списком. Применить шаблон класса для решения задачи № 6 (лаб. работа № 3).

7. Создать шаблон класса для работы с односвязным списком. Применить шаблон класса для решения задачи № 7 (лаб. работа № 3).

8. Создать шаблон класса для работы с бинарным деревом. Применить его для сортировки действительных чисел и строк, вводимых клавиатуры или из файла.

9. Создать шаблон класса **Set** (множество) мощности n (см. задачу № 9, лаб. работа № 2) или создать шаблон класса для работы с односвязным списком. Применить шаблон класса для решения задачи № 9 (лаб. работа № 3).

10. Создать шаблон класса для работы с односвязным списком. Применить шаблон класса для решения задачи № 10 (лаб. работа № 3).

11. Создать шаблон класса, обеспечивающего описание матрицы заданного размера n m и любого минора в ней (см. задачу № 11, лаб. работа № 2) или создать шаблон класса для работы с двусвязным списком. Применить шаблон класса для решения задачи № 11 (лаб. работа № 3).

12. Создать шаблон класса для работы с бинарным деревом. Применить шаблон класса для решения задачи № 12 (лаб. работа № 3).

13. Создать шаблон класса для работы с бинарным деревом. Применить шаблон класса для решения задачи № 13 (лаб. работа № 3).

14. Создать шаблон класса для работы с двусвязным списком. Применить шаблон класса для решения задачи № 15 (лаб. работа № 3).

15. Создать шаблон класса для работы с односвязным списком. Применить шаблон класса для решения задачи № 16 (лаб. работа № 3).

Лабораторная работа №5 Практическая работа №5 Язык C++. Наследование.

При выполнении данной работы необходимо определить базовый класс и производные от него классы. Предусмотреть передачу аргументов конструкторам базового класса; использование виртуальных и перегруженных функций; обработку исключительных ситуаций.

Вариант А

В следующих заданиях требуется создать базовый класс (как вариант абстрактный базовый класс) и определить общие методы `show()`, `get()`, `set()` и другие, специфические для данного класса. Создать производные классы, в которые добавить свойства и методы.

Часть методов переопределить. Создать массив объектов базового класса и заполнить объектами производных классов. Объекты производных классов идентифицировать конструктором по имени или идентификационному номеру.

Вызвать метод `show()` базового класса и просмотреть массив объектов.

Использовать объекты для моделирования реальных ситуаций.

1. Создать базовый класс «Транспортное средство» и производные классы «Автомобиль», «Велосипед», «Повозка». Подсчитать время и стоимость перевозки пассажиров и грузов каждым транспортным средством.

2. Создать базовый класс «Грузоперевозчик» и производные классы «Самолет», «Поезд», «Автомобиль». Определить время и стоимость перевозки для указанных городов и расстояний.

3. Создать аналогичный базовый класс «Пассажироперевозчик» и производные классы «Самолет», «Поезд», «Автомобиль». Определить время и стоимость передвижения.

4. Изменить задания 1–3, чтобы базовый класс стал абстрактным. Сделать некоторые методы абстрактными.

5. Создать базовый класс «Учащийся» и производные классы «Школьник» и «Студент». Создать массив объектов базового класса и заполнить этот массив объектами. Показать отдельно студентов и школьников.

6. Создать базовый класс «Музыкальный инструмент» и производные классы «Ударный», «Струнный», «Духовой». Создать массив объектов «Оркестр». Выдать состав оркестра, переопределив метод.

7. Определить базовый класс «Множество» и производный класс «Кольцо» (операции сложения и умножения обе коммутативные и ассоциативные, связанные законом дистрибутивности; сложение обладает обратной операцией – вычитанием). Ввести кольца целых чисел, многочленов, систему классов целых чисел, сравнимых по модулю. Кольцо является полем, если в нем определена операция деления, кроме деления на нуль. Рациональные числа, дробно-рациональные функции.

8. Создать абстрактный класс «Работник фирмы» и производные классы «Менеджер», «Администратор», «Программист».

9. Создать базовый класс «Домашнее животное» и производные классы «Собака», «Кошка», «Попугай» и др. С помощью конструктора установить имя каждого животного и его характеристики.

10. Создать базовый класс «Садовое дерево» и производные классы «Яблоня», «Вишня», «Груша» и др. С помощью конструктора автоматически установить номер каждого дерева. Принять решение о пересадке каждого дерева в зависимости от возраста и плодоношения.

Вариант Б

1. Создать класс Item (единица хранения в библиотеке), содержащий данные-члены: invNumber – инвентарный номер и taken – взято на руки или имеется в наличии, а также методы: virtual void Show(); //показать информацию о единице хранения bool isAvailable(); // есть ли единица хранения в наличии?

int GetinvNumber(); //возвращает инвентарный номер void Take(); // операция «взять»
void Return(); // операция «вернуть»

Построить производные классы Book и Magazin. Класс Book содержит данные-члены: author, title, publisher, year и методы: Author(); Title(); Publisher(); YearOf Publishing(); Show().

Класс Magazin включает данные-члены: volume; number; year; title
иметоды: Volume(); Title(); Number(); Year(); Show().

2. Создать базовый класс Polygon (многоугольник). Класс должен содержать методы для рисования многоугольника, вычисления периметра, нахождения площади и др. Построить производный класс Triangle (треугольник), содержащий также методы для нахождения точки пересечения медиан, длин медиан, длин биссектрис, координат точек пересечения биссектрис, высот треугольника.

3. Создать абстрактный класс Shape для рисования плоских фигур. Построить производные классы Square (квадрат, который характеризуется координатами левого верхнего угла и длиной стороны), Circle (окружность с заданными координатами центра и радиусом), Ellipse (эллипс с заданными координатами вершин описанного вокруг него прямоугольника), позволяющие рисовать указанные фигуры, а также передвигать их на плоскости.

4. Создать класс CPoint – точка и производные от него классы CcoloredPoint и CLine. На основе классов CcoloredPoint и CLine создать класс CcoloredLine. Все классы должны иметь методы для установки и получения значений всех координат, а также изменения цвета и получения текущего цвета.

5. Описать базовый класс Stroka. Обязательные данные-члены класса: указатель типа char – для хранения строки; значение типа int – длина строки.

Методы: конструктор без параметров; конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом); конструктор копирования; получение длины строки; очистка строки (сделать строку пустой); деструктор.

Описать производный класс «БИТОВАЯ_СТРОКА» (строки данного класса могут содержать только символы '0' и '1'). Если в основе инициализирующей строки встретятся любые символы, отличные от допустимых, то БИТОВАЯ_СТРОКА становится пустой. Содержимое строки рассматривается как двоичное представление целого числа со знаковым разрядом. Отрицательные числа хранятся в дополнительном коде.

Обязательные методы: конструктор без параметров; конструктор, принимающий в качестве параметра C-строку; конструктор копирования; деструктор; изменение знака числа (перевод числа в дополнительный код).

Переопределить следующие операции (длина строки результата в случае необходимости расширяется влево знаковым разрядом): присваивание; сложение (+); проверка на равенство (==).

6. Создать производный класс «СТРОКА10» (целое неотрицательное десятичное число) от класса «СТРОКА» (описание приведено выше).

Методы: конструктор без параметров; конструктор, принимающий в качестве параметра C-строку; конструктор копирования; деструктор; метод, определяющий, можно ли представить данное число в формате int; метод, определяющий, равно ли число нулю; метод, возвращающий представление числа в виде целого (int); метод, удаляющий незначащие нули.

Переопределить операции: сложение (+); проверка на больше (по значению) (>); проверка на меньше (<); присваивание (=).

7. Создать производный класс «БУЛЕВ ВЕКТОР» (**BoolVector**) от класса **Vector**. Компоненты принимают значения из множества {0,1}.

Методы: конструктор без параметров; конструктор, принимающий в качестве параметров указатель на массив целого типа (если элементы массива содержат числа, отличные от 0 и 1, то создается пустой вектор) и размер вектора; конструктор копирования; деструктор; метод, возвращающий число единиц в векторе; метод, возвращающий позицию самой левой единицы в векторе.

Переопределить операции: поразрядная конъюнкция (&); поразрядная дизъюнкция (|); поразрядная инверсия (~); поразрядная операция ИСКЛЮЧАЮЩЕЕ ИЛИ (^); присваивание (=).

8. Создать производный класс «ТРОИЧНЫЙ ВЕКТОР» от класса **Vector**. Компоненты принимают значения из множества {0,1,2}.

Методы: конструктор без параметров; конструктор, принимающий в качестве параметров указатель на массив целого типа и размер вектора; конструктор копирования; деструктор; проверка двух векторов на ортогональность (два троичных вектора называются ортогональными, если в них существует пара одноименных компонент, имеющих в одном из векторов значение 0, а в другом – 1); метод, возвращающий число компонент в векторе, принимающих значение 2.

Переопределить операции: присваивание (=); поразрядная конъюнкция (пересечение) двух не ортогональных векторов (&); $0&0=0$, $1&1=1$, $2&2=2$, $0&2=0$, $2&0=0$, $1&2=1$, $2&1=1$; индексирование ([]).

9. Создать производный класс «БУЛЕВА МАТРИЦА» от класса «ЦЕЛОЧИСЛЕННАЯ МАТРИЦА».

Методы: конструктор без параметров; конструктор, принимающий в качестве параметров целочисленный двумерный массив, содержащий матрицу, и ее размеры n и m ; конструктор копирования; деструктор; метод возвращает число единиц в матрице; метод, возвращающий -каноническую матрицу (в исходной матрице удалены повторяющиеся строки; строки составлены в порядке возрастания неотрицательных чисел, в качестве двоичных кодов которых рассматриваются данные строки).

Переопределить операции: поэлементная конъюнкция двух матриц (&); поэлементная дизъюнкция двух матриц (|); поэлементная операция ИСКЛЮЧАЮЩЕЕ ИЛИ двух матриц (^); произведение двух матриц $A=[a_{ij}]$ и $B=[b_{jk}]$, где $i=1..n$, $j=1..m$, $k=1..l$ (*). При вычислениях операция целочисленного умножения заменяется конъюнкцией, а сложение – дизъюнкцией; присваивание (=).

10. Создать производный класс «МИНОР» для базового класса «МАТРИЦА» размерности n и m . Переопределить для производного класса операции и методы (см. лаб. работу № 2, задание № 4).

11. Расширить возможности стандартного класса Time, чтобы можно было выводить время дня: утро, вечер и т. д.

12. Расширить возможности стандартного класса Date, чтобы можно было выводить время года: зима, лето и т. д.

13. Расширить возможности класса Annotation, чтобы можно было выводить время и дату изменения аннотации.

14. Расширить возможности класса Dictionary, чтобы можно было выводить дату последнего изменения в словаре.

15. Расширить возможности класса File, чтобы можно было выводить время и дату создания файла.

16. Расширить возможности класса Stack, чтобы можно было выводить время последнего сеанса работы со стекком.

17. Определить базовый класс для работы с прямоугольными матрицами, предусмотрев ввод-вывод матриц и выполнение следующих операций: сложение матриц; умножение матрицы на скаляр; перестановка строк матрицы по заданному вектору транспозиции; перестановка столбцов матрицы по заданному вектору транспозиции. В производном классе реализовать указанные операции для квадратных матриц, добавив выполнение следующих операций: транспонирование матрицы; умножение матриц.

Лабораторная работа № 6,7,8

Практическая работа №6,7,8

Язык C++. Виртуальные функции. Язык C++. Поток и файлы. Язык C++.

Многофайловые программы.

При выполнении приводимых ниже заданий можно использовать классы, разработанные в лабораторных работах № 1–3. Осуществлять контроль состояния потоков. В случае возникновения ошибок потоков генерировать и обрабатывать исключительные ситуации. Для соответствующих классов перегрузить операции вставки в поток и извлечения из потока. При динамическом выделении памяти предусмотреть обработку исключения, возникающего при нехватке памяти.

а) Для класса **Student** (лаб. работа № 1) предусмотреть ввод данных из файла. Полученные при выполнении лаб. работы № 1 списки студентов вывести в файл.

То же задание для классов: б) **Abiturient** (лаб. работа № 1); в) **Acroflot** (лаб. работа № 1); г) **Worker** (лаб. работа № 1); д) **Train** (лаб. работа № 1); е) **Product** (лаб. работа № 1);

ж) **Patient** (лаб. работа № 1); з) **Bus** (лаб. работа № 1); и) **Customer** (лаб. работа № 1); к) **File** (лаб. работа № 1); л) **Word** (лаб. работа № 1); м) **House** (лаб. работа № 1); н) **Phone** (лаб. работа № 1);

о) **Person** (лаб. работа № 1).

а) При выполнении задания № 1 лаб. работы № 2 (класс **Complex**) предусмотреть формирование массива объектов путем считывания комплексных чисел из файла. Результат также вывести в файл.

То же задание для классов: б) **Fraction** (лаб. работа № 2); в) **Vector** (лаб. работа № 2). Предусмотреть обработку исключения при динамическом выделении памяти; г) **Matrix** (лаб. работа № 2); д) **Polynom** (лаб. работа № 2); е) **Stack** (лаб. работа № 2); ж) **Строка** (лаб. работа № 2); з) **Set** (лаб. работа № 2);

и) «**Массив строк**» (зад. № 10 лаб. работы № 2); к) «**Булев вектор**» (лаб. работа № 2); л) «**Тройичный вектор**» (лаб. работа № 2); м) «**Булева матрица**» (лаб. работа № 2).

Те же задания, что и в разделах I и II, но для классов, реализующих работу с динамическими структурами данных (см. лаб. работу № 3).

Рекомендуемая литература

Основная литература

1. Лафоре, Р. Объектно-ориентированное программирование в C++ [Текст] / Р. Лафоре. - 4-е изд. - Санкт-Петербург : Питер, 2014. - 928 с. : ил. - (Классика Computer Science) - ISBN 978-5-496-00353-7. (20)

2. Васильев, А. Н. Java. Объектно-ориентированное программирование [Текст] : для магистров и бакалавров. Базовый курс по объектно-ориентированному программированию / А. Н. Васильев. - Санкт-Петербург : Питер, 2014. - 400 с. - (Учебное пособие) - ISBN 978-5-496-00044-4. (15)

Дополнительная литература

1. Павловская, Т.А. C++, Объектно-ориентированное программирование: практикум / Павловская, Т.А. . - СПб. : Питер, 2006. - 265с. : ил. (5)

2. Кириос, В.Н. Информатика II. Основы алгоритмизации и программирования на языке C++ : учебно-методическое пособие / В.Н. Кириос ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Эль Контент, 2013. - 160 с. : ил.,табл., схем. - ISBN 978-5-4332-0068-5 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=208651>.

5.3 Периодические издания

1. Журнал «Вестник компьютерных и информационных технологий»
2. Журнал «Информационные технологии и вычислительные системы»
3. Журнал «Стандарты и качество»
4. Журнал «Прикладная информатика»

Периодические издания

1. Журнал «Вестник компьютерных и информационных технологий»
2. Журнал «Информационные технологии и вычислительные системы»
3. Журнал «Стандарты и качество»
4. Журнал «Информатика и вычислительная техника»