



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
Дальневосточный федеральный университет
(ДФУ)

ШКОЛА ЭКОНОМИКИ И МЕНЕДЖМЕНТА

СОГЛАСОВАНО

УТВЕРЖДАЮ

Руководитель ОП

Заведующий кафедрой бизнес-информатики и
экономико-математических методов

 Е.Г. Юрченко

 Ю.Д. Шмидт

« 12 » сентября 2016 г.

« 12 » сентября 2016 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Программирование

Направление подготовки: 38.03.05 «Бизнес-информатика»

Профиль подготовки: «Моделирование и оптимизация бизнес-процессов»

Форма подготовки очная

курс 1,2 семестр 2,3

лекции 54 час.

практические занятия - час.

лабораторные работы 72 час.

в том числе с использованием МАО лек. - / пр. - / лаб. 36 час.

всего часов аудиторной нагрузки 126 час.

в том числе с использованием МАО 36 час.

самостоятельная работа 126 час.

в том числе на подготовку к экзамену 54 час.

контрольные работы (количество) -

курсовая работа / курсовой проект 3 семестр

зачет - семестр

экзамен 2,3 семестр

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования, утвержденного приказом Министерства образования и науки РФ от 11.08.2016 № 1002

Рабочая программа обсуждена на заседании кафедры бизнес-информатики и экономико-математических методов, протокол № 7 от 12 сентября 2016 г.

Заведующий кафедрой: д-р экон. наук, проф. Ю.Д. Шмидт
Составители: канд. физ.-мат. наук, доцент Л.И. Прудникова

I. Рабочая программа пересмотрена на заседании кафедры:

Протокол от « _____ » _____ 20 г. № _____

Заведующий кафедрой _____
(подпись) (и.о. фамилия)

Изменений нет.

II. Рабочая программа пересмотрена на заседании кафедры:

Протокол от « _____ » _____ 20 г. № _____

Заведующий кафедрой _____
(подпись) (и.о. фамилия)

ABSTRACT

Bachelor's degree in 38.03.05 «Business-Informatics».

Study profile's Program "Title" "The modeling and optimization of business processes".

Course title: Programming

Basic part of Block, 7credits.

Instructor:

At the beginning of the course a student should be able to:

- ability to self-organization and self-education;
- the ability to solve standard tasks of professional activity on the basis of information and bibliographic culture using information and communication technologies and taking into account the basic requirements of information security, the ability to determine the general forms and patterns of a particular subject area;
- the ability to publicly present their own and well-known scientific results, the ability to use methods of mathematical and algorithmic modeling in the analysis of management tasks in the scientific and technical sphere, in the economy, business and humanitarian fields.

Learning outcomes:

OPK-3 ability to work with a computer as a means of managing information, working with information from various sources, including in global computer networks

PC-17 the ability to use the basic methods of natural sciences in professional activities for theoretical and experimental research

Course description:

structure of programming languages, operating environment, algorithms, fundamentals of programming technology

Main course literature:

1. Andreeva, T.A. Programming in Pascal [Electronic resource] / Andreeva TA - Electron. text data. - M .: Internet University of Information Technologies (INTUIT), 2016. - 277 c. - Access mode: <http://www.iprbookshop.ru/52215.html>. - EBS "IPRbooks"
2. Kovrizhnyh A.Yu. Basics of algorithmization and programming. Part 1. Tasks and exercises. Workshop [Electronic resource]: teaching aid / Kovrizhnyh A.Yu., Kononchuk EA, Luzina G.E. - Electron. text data. - Ekaterinburg: Ural Federal University, DIA, 2016. - 52 p. - Access mode: <http://www.iprbookshop.ru/68449.html>. - EBS "IPRbooks"
3. Kovrizhnyh A.Yu. Basics of algorithmization and programming. Part 2. Calculated work. Workshop [Electronic resource]: teaching aid / Kovrizhnyh A.Yu., Kononchuk EA, Luzina G.E. - Electron. text data. - Ekaterinburg: Ural Federal University, DIA, DIA, 2016. - 44 p. - Access mode: <http://www.iprbookshop.ru/68450.html>. - EBS "IPRbooks"
4. Borisenko V.V. Basics of programming [Electronic resource] / Borisenko V.V. - Electron. text data. - M .: Internet University of Information Technologies (INTUIT), 2016. - 323 c. - Access mode: <http://www.iprbookshop.ru/52206.html>. - EBS "IPRbooks"
5. Zaitsev, M. G. Modern programming technologies [Electronic resource]: workshop / M. G. Zaitsev. - Electron. text data. - Novosibirsk: Siberian State University of Telecommunications and Informatics, 2008. - 31 p. - 2227-8397. - Access mode: <http://www.iprbookshop.ru/55460.html>

Form of final control: exam

Аннотация к рабочей программе дисциплины «Программирование»

Учебный курс «Программирование» предназначен для студентов направления подготовки 38.03.05 Бизнес-информатика, профиль «Моделирование и оптимизация бизнес-процессов».

Дисциплина «Программирование» включена в состав базовой части блока «Дисциплины (модули)».

Общая трудоемкость дисциплины составляет 7 зачетных единиц, 252 часа. Учебным планом предусмотрены лекционные занятия (54 часа), лабораторные работы (72 часа, в том числе МАО 36 часов), самостоятельная работа (126 часов, в том числе на подготовку к экзамену 54 часа). Дисциплина реализуется на 1 и 2 курсе во 2-3 семестрах.

Дисциплина «Программирование» основывается на знаниях, умениях и навыках, полученных в результате изучения дисциплин «Теоретические основы информатики», «Приложения линейной алгебры в экономике», «Применение дискретных методов и алгоритмов в экономике», «Численный анализ» и позволяет подготовить студента к освоению ряда таких дисциплин, как «Объектно–ориентированный анализ и программирование», «Стандартизация программного обеспечения», «Базы данных».

Содержание дисциплины «Программирование» охватывает следующий круг вопросов: структуру языков программирования, операционную среду, алгоритмы, основы технологии программирования.

Цель – изучение основных алгоритмов, структур данных и новейшим направлениям в области программирования, основам технологии программирования, освоению разных стилей программирования, выработке умения использовать современные информационные технологии.

Задачи:

- Усвоение и закрепление основных приемов, методов и принципов работы при решении задач на ЭВМ с использованием языка высокого уровня (Паскаль).

- Научить использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования.

- Обучить основными методами, способами и средствами получения, хранения, переработки информации; навыками работы с компьютером как средством управления информацией; языкам программирования.

Для успешного изучения дисциплины «Программирование» у обучающихся должны быть сформированы следующие предварительные компетенции:

- способность к самоорганизации и самообразованию;
- способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности, способностью к определению общих форм и закономерностей отдельной предметной области;
- способность публично представлять собственные и известные научные результаты, способностью использовать методы математического и алгоритмического моделирования при анализе управленческих задач в научно-технической сфере, в экономике, бизнесе и гуманитарных областях.

В результате изучения данной дисциплины у обучающихся формируются следующие общепрофессиональные и профессиональные компетенции (элементы компетенций).

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-3 способностью работать с компьютером как средством управления информацией, работать с	Знает	сущность и значение информации в развитии современного общества; методы разработки программ; системы программирования; современные среды разработки.

информацией из различных источников, в том числе в глобальных компьютерных сетях	Умеет	применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы.
	Владеет	навыками работы с компьютером как средством управления информацией; языками программирования.
ПК-17 способность использовать основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования	Знает	основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования
	Умеет	использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования;
	Владеет	основными методами, способами и средствами получения, хранения, переработки информации.

Для формирования вышеуказанных компетенций в рамках дисциплины «Программирование» применяются следующие методы активного/интерактивного обучения: метод новых вариантов, лекция-консультация, лекция-пресс-конференция.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Раздел.1. Базовые понятия (10 ч.)

Тема 1. Введение

Место дисциплины «Программирование». Формальные средства построения модели, возникающих в предметной области. Объектно-ориентированная методология.

Тема 2. Объектно-ориентированный анализ

Роль анализа в процессе разработки программного обеспечения. Основные понятия объектно-ориентированного анализа. Основные средства анализа и моделирования предметной области в языке UML.

Раздел.2. Объектное программирование на СИ++(20ч.)

Тема 1. Программа на языке СИ++

Лексемы языка СИ++. Функция и структура программы на языке Си++. Синтаксис и семантика операторов языка Си++. Типы данных.

Тема 2. Синтаксис и семантика языка СИ++

Преобразование типов данных. Массивы и строковые данные. Поразрядные логические операции и операции сдвигов. Логические выражения. Передача параметров. Прототипы функций в СИ++. Шаблоны функций. Перегрузка функций. Виртуальные функции. Ссылки. Указатели и операции для них. Классы памяти и области видимости имен. Сложные объявления. Перечислимый тип. Ввод операций в СИ++.

Тема 3. Данные типа структура в языке СИ++ (5ч.)

Структура в языке СИ++. Списки, очереди, стеки, деревья и их реализация.

Раздел.3. Объектно-ориентированное проектирование и программирование(ООП)(24ч.)

Тема 1. Основные принципы

Основные принципы и компоненты ООП. Объект и класс. Отношения между объектами, классами. Стили программирования. Парадигма ООП.

Тема 2. Шаблоны

Область видимости. Наследование. Шаблоны классов. Пример системы спроектированной и запрограммированной в соответствии с ООП.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА Лабораторные работы (72ч.)

Лабораторная работа №1.

Простейший тип программы(8ч.)

Лабораторная работа №2.

Указатели(12ч.)

Лабораторная работа №3.

Абстрактные типы данных(20ч.)

Лабораторная работа №4.

Объектно-ориентированное программирование(32ч.)

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Программирование» представлено в Приложении 1 и включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1	Раздел.1. Базовые понятия .	ОПК-3 ПК-17	Знает сущность и значение информации в развитии современного общества; методы разработки программ; системы программирования; современные среды разработки.	Конспект(ПР-7) ПР-1 тест	Вопросы к экзамену 1-10

2.	Раздел.2. Объектное программирование на СИ++	ОПК-3 ПК-17	Знает основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования	Конспект(ПР-7) ПР-1 тест	Вопросы к экзамену 11-18
3.	Раздел.3. Объектно-ориентированное проектирование и программирование(ООП)(15ч.).	ОПК-3 ПК-17	Умеет применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы.	Конспект(ПР-7) ПР-1 тест	Вопросы к экзамену 19-25
4.	Лабораторные работы №1-4	ОПК-3 ПК-17	Владеет навыками работы с компьютером как средством управления информацией; языками программирования.	Лабораторная работа (ПР-6)	Отчет

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

1. Андреева Т.А. Программирование на языке Pascal [Электронный ресурс]/ Андреева Т.А. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 277 с. – Режим доступа: <http://www.iprbookshop.ru/52215.html>. – ЭБС «IPRbooks»
2. Коврижных А.Ю. Основы алгоритмизации и программирования. Часть 1. Задачи и упражнения. Практикум [Электронный ресурс]: учебно-методическое пособие/ Коврижных А.Ю., Конончук Е.А., Лузина Г.Е. – Электрон. текстовые данные. – Екатеринбург: Уральский федеральный

- университет, ЭБС АСВ, 2016. – 52 с. – Режим доступа: <http://www.iprbookshop.ru/68449.html>. – ЭБС «IPRbooks»
3. Коврижных А.Ю. Основы алгоритмизации и программирования. Часть 2. Расчетные работы. Практикум [Электронный ресурс]: учебно-методическое пособие/ Коврижных А.Ю., Конончук Е.А., Лузина Г.Е. – Электрон. текстовые данные. – Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016. – 44 с. – Режим доступа: <http://www.iprbookshop.ru/68450.html>. – ЭБС «IPRbooks»
4. Борисенко В.В. Основы программирования [Электронный ресурс]/ Борисенко В.В. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 323 с. – Режим доступа: <http://www.iprbookshop.ru/52206.html>. – ЭБС «IPRbooks»
5. Зайцев, М. Г. Современные технологии программирования [Электронный ресурс] : практикум / М. Г. Зайцев. — Электрон. текстовые данные. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2008. — 31 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/55460.html>

Дополнительная литература

1. Харитонов, Е. А. Основы программирования для студентов технологического профиля [Электронный ресурс] : учебное пособие / Е. А. Харитонов, А. К. Сафиуллина. — Электрон. текстовые данные. — Казань : Казанский национальный исследовательский технологический университет, 2014. — 151 с. — 978-5-7882-1558-7. — Режим доступа: <http://www.iprbookshop.ru/63754.html>
2. Хвощев, С. В. Основы программирования в Delphi для ОС Android [Электронный ресурс] / С. В. Хвощев. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных

- Технологий (ИНТУИТ), 2016. — 85 с. — 2227-8397. — Режим доступа:
<http://www.iprbookshop.ru/73694.html>
3. Борисенко, В. В. Основы программирования [Электронный ресурс] / В. В. Борисенко. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 323 с. — 978-5-9556-00039-0. — Режим доступа:
<http://www.iprbookshop.ru/52206.html>
4. Курипта, О. В. Основы программирования и алгоритмизации [Электронный ресурс] : практикум / О. В. Курипта, О. В. Минакова, Д. К. Проскурин. — Электрон. текстовые данные. — Воронеж : Воронежский государственный архитектурно-строительный университет, ЭБС АСВ, 2015. — 133 с. — 978-5-89040-575-3. — Режим доступа: <http://www.iprbookshop.ru/59123.html>
5. Программирование на Языке Pascal. Видеокурс. — Режим доступа: <https://www.intuit.ru/studies/courses/41/41/info>

Текущие публикации в следующих изданиях:

1. Журнал «Программирование» РАН
2. Журнал «Пуск»
3. Журнал «Компоненты и технологии»
4. Журнал «Мир ПК»
5. Журнал «ПРОграммист»

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Единое окно доступа к образовательным ресурсам - <http://window.edu.ru/library>
2. Электронно - библиотечная система образовательных и просветительских изданий - <http://www.iqlib.ru/>

3. Электронно-библиотечная система. Электронные версии учебной литературы по естественным, техническим и гуманитарным наукам -
<http://www.twirpx.com/file>,
<http://www.twirpx.com/files/informatics/languages/pascal/>,
<http://www.twirpx.com/library/comp/algorithms/>,
<http://www.twirpx.com/library/comp/cpp/>
4. Электронная библиотека издательства «Юрайт» www.biblio-online.ru
5. Научная библиотека ДВФУ. Электронный каталог <http://lib.dvfu.ru:8080/>
6. Информационный портал. Все о компьютере и программировании для начинающих <http://info-comp.ru/knigigutnalifile/>

Электронные ресурсы:

1. Школа программирования <http://prog-school.ru/>
2. Форум программистов <http://www.hardforum.ru/f22/>
3. Библиотека разработчика <http://www.libray.narod.ru/>
4. ВІВлиотека программиста <http://bib.com.ua/>

Перечень информационных технологий и программного обеспечения

Лекции проводятся с использованием проектора и мультимедийного комплекса для проведения лекций внутренней системы портала ДВФУ. Лабораторные занятия проводятся в специализированном компьютерном классе. Для составления документации используется текстовый процессор (LibreOffice или MicrosoftWord).

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Реализация дисциплины «Программирование» предусматривает следующие виды учебной работы: лекции, практические занятия, самостоятельную работу студентов, текущий контроль и промежуточную аттестацию.

Освоение курса дисциплины «Программирование» предполагает рейтинговую систему оценки знаний студентов и предусматривает со стороны преподавателя текущий контроль за подготовкой и выполнением всех видов работ с обязательным предоставлением отчета о работе, выполнением всех видов самостоятельной работы.

Промежуточной аттестацией по дисциплине «Программирование» является --.

В течение учебного семестра обучающимся нужно:

- изучить теоретический материал (15 баллов);
- успешно выполнить аудиторные и контрольные задания (60 баллов);
- своевременно и успешно выполнить самостоятельные работы (20 баллов).

Студент считается аттестованным по дисциплине «Программирование» при условии выполнения всех видов текущего контроля и самостоятельной работы, предусмотренных учебной программой.

Критерии оценки по дисциплине «Общая теория систем и системный анализ» для аттестации на экзамене следующие: 86-100 баллов – «отлично», 76-85 баллов – «хорошо», 61-75 баллов – «удовлетворительно», 60 и менее баллов – «неудовлетворительно».

Пересчет баллов по текущему контролю и самостоятельной работе производится по формуле:

$$P(n) = \sum_{i=1}^m \left[\frac{O_i}{O_i^{max}} \times \frac{k_i}{W} \right],$$

где: $W = \sum_{i=1}^n k_i^n$ для текущего рейтинга;

$W = \sum_{i=1}^m k_i^n$ для итогового рейтинга;

$P(n)$ – рейтинг студента;

m – общее количество контрольных мероприятий;

n – количество проведенных контрольных мероприятий;

O_i – балл, полученный студентом на i -ом контрольном мероприятии;

O_i^{max} – максимально возможный балл студента по i -му контрольному мероприятию;

k_i – весовой коэффициент i -го контрольного мероприятия;

k_i^n – весовой коэффициент i -го контрольного мероприятия, если оно является основным, или 0, если оно является дополнительным.

Дисциплина изучается в следующих организационных формах: лекции, практические занятия, лабораторное занятие; самостоятельное изучение теоретического материала; самостоятельное выполнение индивидуального задания; индивидуальные консультации.

Работа на лекции

Слушание и запись лекций – сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность студента. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим студентом. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Запись лекций рекомендуется вести по возможности собственными формулировками. Желательно запись осуществлять на одной странице, а следующую оставлять для проработки учебного материала самостоятельно в домашних условиях. Конспект лекции лучше подразделять на пункты, параграфы, соблюдая красную строку. Этому в большой степени будут способствовать пункты плана лекции, предложенные преподавателям. Принципиальные места, определения, формулы и другое следует сопровождать замечаниями «важно», «особо важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек. Лучше если они будут собственными, чтобы не приходилось просить их у однокурсников и тем

самым не отвлекать их во время лекции. Целесообразно разработать собственную «маркографию» (значки, символы), сокращения слов. Не лишним будет и изучение основ стенографии. Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть знаниями.

Вопросы для самоконтроля по разделу 1

1. Структура программы на языке СИ.
2. Синтаксис и семантика операторов действия, элементарных и сложных.
3. Операции присваивания и запятая.
4. Арифметические типы и выражения. Арифметические операции.
5. Представление информации в ЭВМ.
6. Массивы в языке СИ. Строковые данные.
7. Константы литерные и арифметические.
8. Побитовые операции и моделирование работы с множеством.
9. Поразрядные логические операции и операции сдвигов. Примеры.

Задача.

10. Логические выражения.

Вопросы для самоконтроля по разделу 2

11. Указатели. Операции над ними.
12. Функция в языке СИ. Передача параметров. Передача функции как параметра.
13. Структура в языке СИ. Примеры.
14. Списки, очереди, стеки, деревья и их реализация. ПОЗ.
15. Классы памяти и области видимости переменных и функций.
16. Синтаксис описания и объявления в языке СИ. Сложные объявления.

17. Прототипы функций в СИ++. Шаблоны функций. Перегрузка функций.

18. Ссылки.

Вопросы для самоконтроля по разделу 3

19. Объектно-ориентированное проектирование и программирование.

Основные принципы. Основные компоненты.

20. Объект и его свойства. Операции над объектами. Отношения между объектами.

21. Класс. Отношения между классами.

22. Объект и класс в СИ++. Область видимости. Наследование. Шаблоны классов. Пример программы.

23. Константы. Перечислимый тип. Ввод операций в СИ++.

24. Стили программирования и методология ООП.

25. Парадигма языка UML.

Лабораторные работы

При выполнении лабораторной работы необходимо следовать методическим рекомендациям по ее выполнению. Результатом лабораторной работы является созданный документ, который демонстрируется преподавателю в конце работы. Студент должен уметь отвечать на вопросы преподавателя, поясняя процесс создания документа и выполнения работы.

Самостоятельная работа студента

Основной формой самостоятельной работы студента является изучение теоретического материала, его дополнение рекомендованной литературой, выполнение индивидуальных заданий, а также активная работа на практических и лабораторных занятиях. Целью выполнения индивидуальных заданий является закрепление практических навыков, полученных в процессе выполнения лабораторной работы.

Контроль за выполнением самостоятельной работы студента производится в виде контроля каждого этапа работы (см. приложение 1). Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

VIII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лекции проводятся с использованием проектора и внутренней системы портала ДВФУ. Лабораторные занятия проходят в аудиториях, оборудованных компьютерами типа Lenovo C360G-i34164G500UDK с лицензионными программами MicrosoftOffice 2013 и аудио-визуальными средствами проектор Panasonic DLPPjectorPT-D2110XE, плазма LG FLATRON M4716CCBAM4716CJ. Для выполнения самостоятельной работы студенты в жилых корпусах ДВФУ обеспечены Wi-Fi.

В читальных залах Научной библиотеки ДВФУ предусмотрены рабочие места для людей с ограниченными возможностями здоровья, оснащены дисплеями и принтерами Брайля; оборудованные портативными устройствами для чтения плоскочечатных текстов, сканирующими и читающими машинами, видеоувелечителем с возможностью регуляции цветовых спектров; увеличивающими электронными лупами и ультразвуковыми маркировщиками.

В целях обеспечения специальных условий обучения инвалидов и лиц с ограниченными возможностями здоровья ДВФУ все здания оборудованы пандусами, лифтами, подъемниками, специализированными местами, оснащенными туалетными комнатами, табличками информационно-навигационной системы.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЭКОНОМИКИ И МЕНЕДЖМЕНТА

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ
РАБОТЫ ОБУЧАЮЩИХСЯ**

по дисциплине «Программирование»
Направление подготовки: 38.03.05, Бизнес-информатика
Образовательная программа «Моделирование и оптимизация бизнес-процессов»
Форма подготовки (очная)

**Владивосток
2016**

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1.	1-4-я недели семестра	Выбор темы ИПС №1. Разработка алгоритма. Разработка ПО для ИПС.	18 ч.	Утверждение в ходе лабораторной работы Опрос (анализ и обсуждение)
2.	5-8-я недели семестра	Выбор темы ИПС №2. Разработка алгоритма. Разработка ПО для ИПС.	18 ч.	Утверждение в ходе лабораторной работы Опрос (анализ и обсуждение)
3.	9-13-я недели семестра	Выбор темы ИПС №3. Разработка алгоритма. Разработка ПО для ИПС.	18 ч.	Утверждение в ходе лабораторной работы Опрос (анализ и обсуждение)
4.	14-18-я недели семестра	Выбор темы ИПС №4. Разработка алгоритма. Разработка ПО для ИПС.	18 ч.	Утверждение в ходе лабораторной работы Опрос (анализ и обсуждение)
5.	19 неделя семестра	Подготовка к экзамену	54 ч.	экзамен
	итого		126 ч.	

Рекомендации по самостоятельной работе студентов

Самостоятельная работа студентов состоит в работе с литературой, подготовке к лабораторным работам и выполнении индивидуальных заданий по темам.

Работа с литературой

В процессе подготовки к лабораторным работам студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами Интернета является наиболее эффективным методом получения знаний,

позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме.

Подготовка к лабораторным работам

Подготовку к каждой лабораторной работе каждый студент должен начать с изучения теоретического материала и ознакомления с имеющимися реализациями.

Индивидуальные задания

По теме лабораторной работы студентам выдаются индивидуальные задания, предназначенные для закрепления умений и навыков, полученных при выполнении лабораторной работы. Задания выполняются самостоятельно. По каждому заданию предполагается подготовка индивидуального документа (проекта). Выполненный проект демонстрируется преподавателю в начале следующей лабораторной работы. Структура индивидуального задания определяется темой лабораторной работы.

Критерии оценки индивидуальных заданий (проектов)

– 100-86 баллов выставляется, если содержание и составляющие части соответствуют выданному заданию. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 85-76 - баллов выставляется, если при выполнении задания допущено не более одной ошибки. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 75-61 балл выставляется, если при выполнении задания допущено не более двух ошибок. Продемонстрировано знание и владение навыками подготовки документа по теме. Допущено не более 2 ошибок, связанных с пониманием структуры и содержания задания.

– 60-50 баллов - если структура и содержание задания не соответствуют требуемым.

Шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЭКОНОМИКИ И МЕНЕДЖМЕНТА

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине «Программирование»
Направление подготовки: 38.03.05, Бизнес-информатика
Образовательная программа «Моделирование и оптимизация бизнес-процессов»
Форма подготовки (очная)

Владивосток
2016

Паспорт ФОС

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-3 способностью работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях	Знает	сущность и значение информации в развитии современного общества; методы разработки программ; системы программирования; современные среды разработки.
	Умеет	применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы.
	Владеет	навыками работы с компьютером как средством управления информацией; языками программирования.
ПК-17 способность использовать основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования	Знает	основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования
	Умеет	использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования;
	Владеет	основными методами, способами и средствами получения, хранения, переработки информации.

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
1	Раздел.1. Базовые понятия .	ОПК-3 ПК-17	Знает сущность и значение информации в развитии современного общества; методы разработки программ; системы программирования; современные среды разработки.	Конспект(ПР-7) ПР-1 тест	Вопросы к экзамену 1-10
2.	Раздел.2. Объектное программирование на СИ++	ОПК-3 ПК-17	Знает основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования	Конспект(ПР-7) ПР-1 тест	Вопросы к экзамену 11-18

3.	Раздел.3. Объектно-ориентированное проектирование и программирование(ООП)(15ч.).	ОПК-3 ПК-17	Умеет применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы.	Конспект(ПР-7) ПР-1 тест	Вопросы к экзамену 19-25
4.	Лабораторные работы №1-4	ОПК-3 ПК-17	Владеет навыками работы с компьютером как средством управления информацией; языками программирования.	Лабораторная работа (ПР-6)	Отчет

Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
ПК-17 способность использовать основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования	знает (пороговый уровень)	основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования	Знание функционально-алгоритмической структуры разрабатываемого программного продукта	Способность применять информационные технологии на рабочем месте
	умеет (продвинутый)	использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования;	Умение осуществлять анализ функционирования проекта	Способность перечислить принципы организации проектирования и содержание этапов процесса разработки программных комплексов;
	владеет (высокий)	основными методами, способами и средствами получения, хранения, переработки информации.	Владение технологией построения простейших распределенных информационных систем и обеспечения безопасности.	Способность показать навыки уверенного владения средствами поиска информации в Internet, в различных источниках, рекомендованных для самостоятельного изучения.
ОПК-3 Способность работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях	знает (пороговый уровень)	методы структурного и объектно-ориентированного программирования;	Знание базовых шаблонов проектирования - особенностей построения объектно-ориентированных программных систем на C++.	Способность применять полученные знания для решения новых задач в различных областях.
	умеет (продвинутый)	применять к решению прикладных задач базовые алгоритмы обработки информации	Умение применять методики использования программных средств для решения	Способность дать четкие определения основных понятий объектно-ориентированного программирования

			практических задач	
	владеет (высокий)	навыками работы с компьютером как средством управления информацией; языками программирования.	Владение навыками построения модели объектов предметной области	Способность демонстрировать умение применять основные технологии программирования, в частности возможности объектно-ориентированного программирования.

Оценочные средства для проверки сформированности компетенций

Код и формулировка компетенции	Задание
ПК-17 способность использовать основные методы естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования	Дана строка символов: до точки. Группы символов в ней между группами пробелов считаются словами. Определить, сколько слов содержат ровно две буквы «о». Составить блок-схему и написать программу.
ОПК-3 Способность работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях	Расстояние между двумя словами равной длины - это количество позиций, в которых различаются эти слова. В заданном предложении найти пару наиболее далеко удалённых слов заданной длины.

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Текущая аттестация студентов. Текущая аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация проводится в форме вопросов к экзамену для проверки теоретических знаний, а также в форме защиты проекта, выполняемого в рамках самостоятельной работы параллельно с лабораторными работами и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- степень усвоения теоретических знаний - оценивается в форме вопросов к зачету;
- уровень овладения практическими умениями и навыками – оценивается в форме защиты индивидуального задания (проекта).

Промежуточная аттестация студентов. Промежуточная аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

По дисциплине предусмотрен экзамен, который проводится в устной форме.

Критерии выставления оценки студенту на экзамене

Баллы (рейтингов ой оценки)	Оценка зачета (стандартная)	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	Оценка «зачтено» / «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
76-85	«зачтено»/ «хорошо»	Оценка «зачтено» / «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
61-75	«зачтено»/ «удовлетворительно»	Оценка «зачтено» / «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
0-60	«не зачтено»/ «неудовлетворительно»	Оценка «не зачтено» / «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

ВОПРОСЫ К ЭКЗАМЕНУ по дисциплине «Программирование»

Раздел.1. Базовые понятия .

1. Структура программы на языке СИ.
2. Синтаксис и семантика операторов действия, элементарных и сложных.

3. Операции присваивания и запятая.
4. Арифметические типы и выражения. Арифметические операции.
5. Представление информации в ЭВМ.
6. Массивы в языке СИ. Строковые данные.
7. Константы литерные и арифметические.
8. Побитовые операции и моделирование работы с множеством.
9. Поразрядные логические операции и операции сдвигов. Примеры.

Задача.

10. Логические выражения.

Раздел.2. Объектное программирование на СИ++.

11. Указатели. Операции над ними.
12. Функция в языке СИ. Передача параметров. Передача функции как параметра.
13. Структура в языке СИ. Примеры.
14. Списки, очереди, стеки, деревья и их реализация. ПОЗ.
15. Классы памяти и области видимости переменных и функций.
16. Синтаксис описания и объявления в языке СИ. Сложные объявления.
17. Прототипы функций в СИ++. Шаблоны функций. Перегрузка функций.
18. Ссылки.

Раздел.3. Объектно-ориентированное проектирование и программирование(ООП)(15ч.)

19. Объектно-ориентированное проектирование и программирование. Основные принципы. Основные компоненты.
20. Объект и его свойства. Операции над объектами. Отношения между объектами.
21. Класс. Отношения между классами.

22. Объект и класс в СИ++. Область видимости. Наследование. Шаблоны классов. Пример программы.
23. Константы. Перечислимый тип. Ввод операций в СИ++.
24. Стили программирования и методология ООП.
25. Парадигма языка UML.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕЙ АТТЕСТАЦИИ ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ

Тема1: *Простейший тип программы.*

ВАРИАНТЫ:

1. Множество точек на плоскости назовем регулярным, если вместе с каждой парой различных точек оно содержит также еще одну, третью, вершину правильного треугольника с вершинами в этих точках. Определить: регулярно ли заданное множество точек.
2. На плоскости n множеств по m точек в каждом. Среди точек первого множества найти такую, которая принадлежит наибольшему количеству множеств.
3. На плоскости заданы множество точек A и множество окружностей B . Найти две различные точки из A такие, чтобы проходящая через них прямая пересекалась с максимальным количеством окружностей B .
4. На плоскости заданы множество точек A и множество прямых B . Найти две такие различные точки из A , чтобы проходящая через них прямая была параллельна наибольшему количеству прямых B .
5. Выбрать три различные точки из заданного множества точек на плоскости так, чтобы была минимальной разность между количествами точек, лежащих внутри и вне треугольника, с вершинами в выбранных точках.
6. Определить радиус и центр окружности, проходящей по крайней мере

через три различные точки заданного множества точек на плоскости и содержащей внутри наибольшее количество точек этого множества.

7. Характеристикой столбца: целочисленной матрицы назовем сумму модулей его отрицательных нечетных элементов. Переставляя столбцы заданной матрицы, расположить их в соответствии с ростом характеристик.
8. Из заданного множества точек на плоскости выбрать три разные точки A, B, C так, чтобы внутри, треугольника ABC содержалось максимальное количество точек этого множества.
9. Из заданного множества, точек на плоскости выбрать две различные точки так, чтобы окружности заданного радиуса с центрами в этих точках содержали внутри себя одинаковое количество заданных точек
10. Среди треугольников с вершинами в заданном множестве точек на плоскости указать такой, стороны которого содержат максимальное число точек заданного множества.
11. Из заданного на плоскости множества точек выбрать три различные точки так, чтобы разность между площадью круга ограниченного окружностью, проходящей через эти три точки, и площадью треугольника, с вершинами в заданных точках, была минимальной.
12. Найти такую точку заданного на плоскости множества точек, сумма расстояний от которой-до остальных минимальна.
13. Среди треугольников с вершинами в заданном множестве точек на плоскости указать такой, стороны которого содержат максимальное число точек заданного множества.
14. Порядок на точках плоскости определим следующим образом: $(x, y) \leq (\mu, \gamma)$ если либо $x < \mu$, либо $x = \mu$ и $y \leq \gamma$. Перечислить точки заданного множества точек на плоскости в соответствии с этим порядком.
15. Характеристикой столбца целочисленной матрицы назовем сумму квадратов модулей его отрицательных элементов. Найти \max характеристику.

16. Дано $3 \cdot n$ точек на плоскости, причем никакие три из них не лежат на одной прямой. Построить множество треугольников с вершинами в этих точках, так, чтобы никакие два треугольника не пересекались и не содержали друг друга.
17. Из заданного множества точек на плоскости выбрать три разные точки A, B, C так, чтобы внутри треугольника ABC содержалось максимальное количество точек этого множества.
18. Из заданного множества точек на плоскости выбрать две различные точки так, чтобы окружности заданного радиуса с центрами в этих точках содержали внутри себя одинаковое количество заданных точек.
19. Даны две непересекающихся конечных множества точек на плоскости. Определить окружность, проходящую через k ($k \geq 3$) точек каждого из множеств.
20. Среди треугольников с вершинами в заданном множестве на плоскости указать такой, стороны которого содержат максимальное число точек заданного множества.
21. Построить множество всех различных выпуклых четырехугольников с вершинами в заданном множестве точек на плоскости.
22. Даны два множества точек на плоскости. Выбрать три различные точки первого множества так, чтобы круг, ограниченный окружностью, проходящей через эти точки, содержал все точки второго множества и имел минимальную площадь.
23. Дана матрица A размером 20×20 . Считая ее составленной из 100 квадратов размером 2×2 и переставляя эти квадраты, преобразовать A так, чтобы в результирующей матрице для всяких двух квадратов B и C выполнялось следующее условие; если сумма элементов B меньше суммы элементов C , то B лежит либо выше, либо левее (когда B и C на одной горизонтали) квадрата C .
24. На плоскости задано множество окружностей. Две окружности A и B назовем связанными, если они пересекаются, либо существует третья

окружность C заданного множества, связанная с A и B . Выбрать максимальное подмножество попарно не связанных друг с другом окружностей.

25. Из заданного на плоскости множества точек выбрать три различные точки так, чтобы разность между площадью круга, ограниченного окружностью, проходящей через эти три точки, и площадью треугольника с вершинами в этих точках была минимальной.

Тема2: *Указатели.*

ВАРИАНТЫ:

1. Написать функцию

Char find_idt()*

которая

а) помещает в *string[0]* содержимое буфера литеры *buf_lit*;

б) из входного потока читает литеру за литерой, пока не встретит не букву и не цифру или конец файла;

в) считанную литеру помещает в очередную позицию буфера строки, если она цифра или буква. Если наступило условие окончания чтения литеры, то в конец строки добавляется символ "0", а литера остается в буфере *buf_lit*. Если длина буфера исчерпана, но на входе - буква или цифра, то чтение продолжается без заполнения буфера строки;

г) после чтения содержимое буфера строки копируется в оперативную память, и адрес начала строки возвращается функцией. Если память для строки не может быть выделена, то функция возвращает *NULL*.

2. Написать функцию

Char find_const()*

которая

а) помещает в буфер строки *digit[0]* литеру *buf_lit*;

б) из входного потока читает литеру за литерой, пока не встретит не цифру или конец файла;

в)если считанная литера – цифра, то она помещает в очередную позицию буфера строки. Если это не цифра, то строка дополняется символом "0",а литера остается в буфере *buf_lit*. Если буфер исчерпан, а на входе - цифра, то чтение продолжается пока не встретится не цифра, но литера в буфер не заносится;

г)после чтения содержимое буфера строки копируется в оперативную память, и адрес начала строки возвращается функцией. Если память для строки не может быть выделена, то функция возвращает *NULL*.

3. Написать функцию

int incl_lexm (char p_lexm, char type_lexm),*

которая методом хеширования для строки, адресуемой *p_lexm*, определяет свободную позицию в таблице *tbl* и, если в ней отсутствует идентичная строка, то включает её адрес в таблицу. Возвращается индекс включения адреса или индекс адреса идентичной строки. Помимо адреса включается тип лексемы: идентификатор или константа (признаки "i" и "c"). Если включается константа, то её значение переводится в число и, если оно по модулю больше $2^{15}-1$ то заменяется на $2^{15}-1$. Полученное число вносится в таблицу. Если адрес нельзя включить в таблицу, то выдаётся сообщение 12: не могу включить "аргумент" в таблицу из-за переполнения.

Метод хеширования:

Из кода каждой литеры удаляются три старших разряда, оставшиеся разряды трех кодов объединяются в 15-ти битовый код и интерпретируются как код двоичного числа. Все двоичные числа, полученные из всех кодов строки, суммируются и делятся с остатком на *VTABL_ELM*. Этот остаток и является стартовым числом при поиске свободного места в таблице.

Если указанная позиция свободна, то выполняется включение, иначе строки проверяются на идентичность. Если строки не идентичны, то следующая позиция отстоит от предыдущей на остаток от деления первого кодового числа на *STEP_HESH*. Этот-процесс повторяется до тех пор, пока

не будет найдена идентичная строка или свободная, или получено вновь стартовое число, что означает переполнение таблицы и невозможность включить в неё новый элемент.

4. Расстояние между двумя словами равной длины - это количество позиций, в которых различаются эти слова. В заданном предложении найти пару наиболее далеко удалённых слов заданной длины.
5. В предложении все слова начинаются с различных букв. Напечатать (если можно) слова предложения в таком порядке, чтобы последняя буква каждого слова совпадала с первой буквой следующего слова.
6. В заданном предложении указать слово, в котором доля гласных (А, Е, У, О) максимальна.
7. Переставить и распечатать слова заданного предложения в соответствии с ростом доли согласных (В, С, Д, Ф, Г, Н, К, Л, М, П, Р, Q, S, Т, V, W, X, Z) в этих словах.
8. Отредактировать заданное предложение, удаляя из него все слова с нечётными номерами и переворачивая слова с чётными номерами.

ПРИМЕР:

HOW DO YOU DO -> OD OD

Тема3: *Абстрактные типы данных .*

ВАРИАНТЫ:

1. В исходном тексте встречаются идентификаторы в русском алфавите. Дана таблица из 256 элементов. Длина элемента не известна. Используя узлы списка, содержащего по 10 байт в каждом, каждый идентификатор разместить в списке и включить указатель на него в таблицу последовательно.

ПРИМЕР:



и т.д.

Рис.51.

- В исходном тексте встречаются целочисленные константы. Написать программу, которая выделяет все константы и располагает их в линейном однородном списке по убыванию. Программа должна иметь функцию печати.
- В исходном тексте встречаются цепочки вида буква цифра буква цифра... Выделить все такие цепочки (длина их не известна) и включить их в таблицу, вначале расположив самые длинные, затем короче. Например, если имеются цепочки вида a1a2в3 a4в6+ a9e7a2a2...

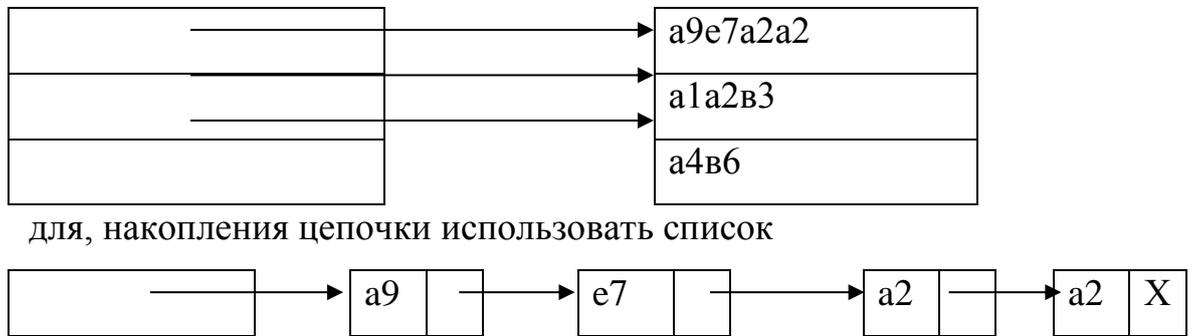


Рис.52.

Для ускорения поиска иметь вспомогательный указатель на последний элемент списка.

- В исходном тексте встречаются всевозможные символы языка Си (+, -, ++, и т.д.).

Написать программу, которая выделяет каждое вхождение отдельного символа, считает их количество и хранит их вместе со счётчиками в линейном однонаправленном циклическом списке.

- Во входном потоке встречаются в произвольном порядке подцепочки alal+alalal * a1 alalal ...

Выделить все такие подцепочки и включить их в таблицу в порядке убывания длин цепочек по схеме

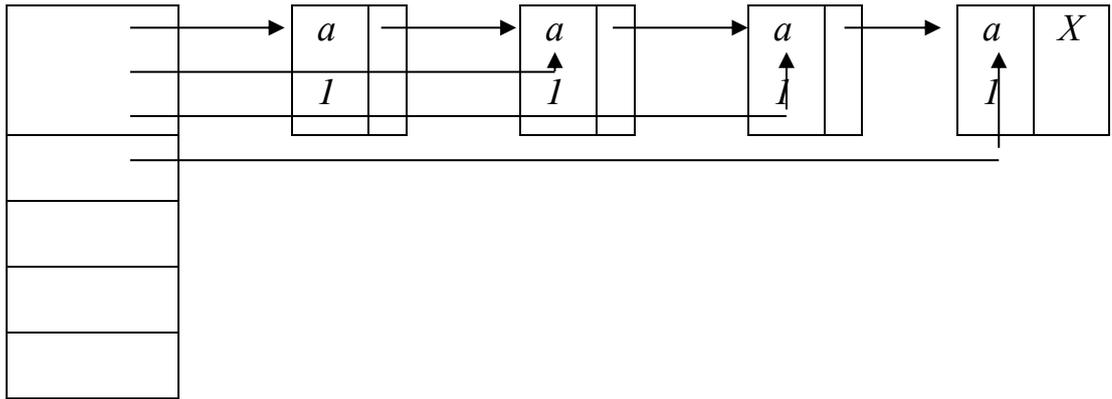


Рис.53.

т.е. в ОП должна присутствовать лишь самая длинная цепочка из символов $a1$ В массиве указателей не более 100 элементов.

В начале создаётся список для первой цепочки, затем он наращивается слева по мере нахождения более длинных цепочек, но указатели на начало сортируются так, что каждый ниже лежащий начинает список той же длины или меньше, чем список, начинаемый выше лежащим указателем.

6. Пусть на входе программы последовательность чисел с плавающей точкой. Используя узел вида:

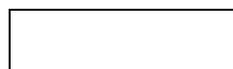
Левый указатель	Число	Правый указатель
--------------------	-------	---------------------

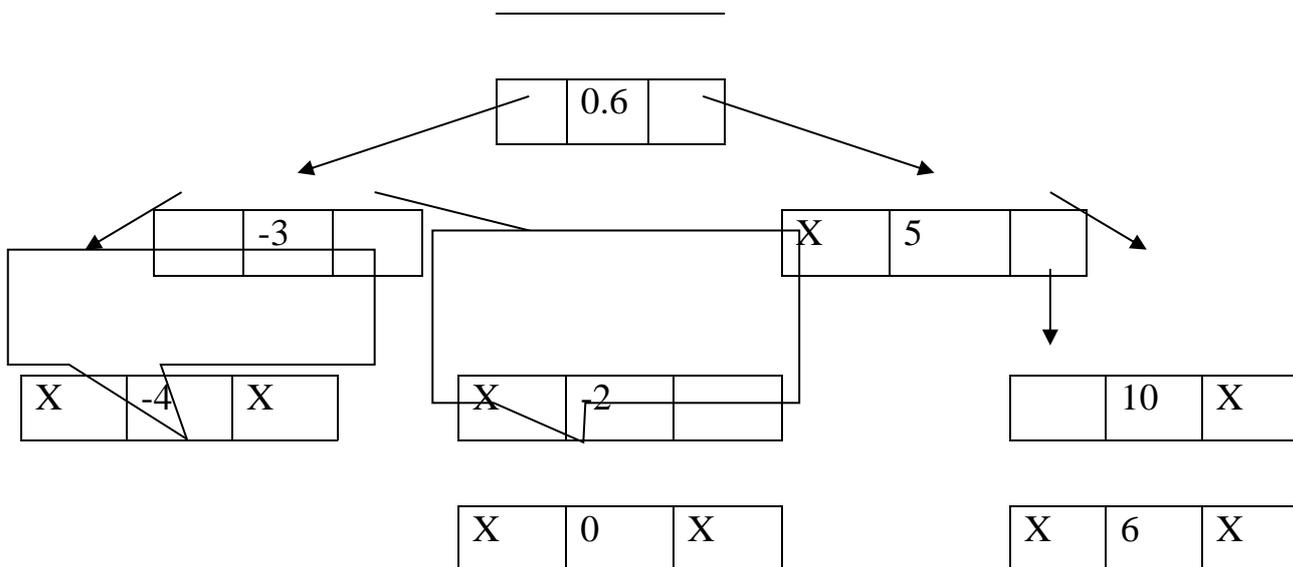
Рис.54.

сформировать список, в котором все узлы левее "некоторого" и ниже его содержат числа меньше числа из "некоторого", и все узлы правее его и ниже содержат числа больше из "Некоторого".

ПРИМЕР:

tree





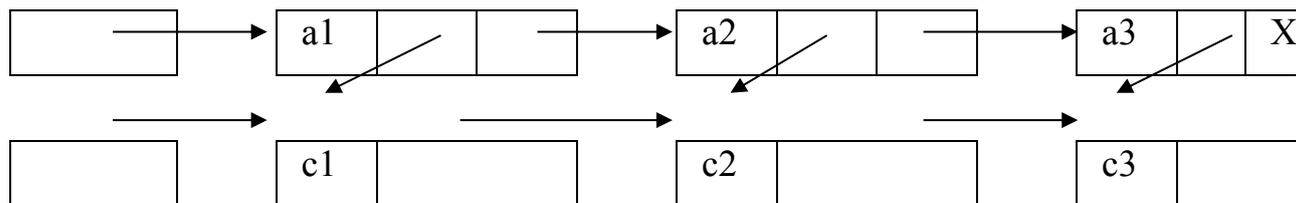
левее и ниже

правее и ниже

Рис.55.

7. Из входного потока в произвольном порядке содержащим одиночные буквы русского языка и константы целых чисел, читаются пары (буква, число);

Если в последней паре не хватает числа, считать его 0. Буква считается одиночной, если слева и справа от нее - не буква. Будем считать, что следом за буквой не цифра. Написать программу, которая формирует два > списка.



Здесь a_i - буква, c_i - число.

Рис.56.

Тема4: Объектно-ориентированное программирование.

ВАРИАНТЫ:

1. Создать класс Point для работы с точками на плоскости. Координаты точки - декартовы. Обязательно должны быть реализованы: перемещение точки по оси X, перемещение точки по оси Y, определение расстояния до начала

координат, расстояние между двумя точками. Программа должна содержать меню, позволяющие осуществить проверку всех методов класса.

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

3. Реализовать класс `Bankomat`, моделирующий работу банкомата. В классе должны содержаться поля для хранения идентификационного номера банкомата, информация о текущей сумме денег, оставшейся в банкомате, минимальных и максимальных суммах, которые позволяют снять клиенту в один день. Реализовать метод инициализации банкомата, метод загрузки купюр в банкомат и метод снятия определенной суммы денег. Метод снятия денег должен выполнять проверку на корректность снимаемой суммы.

4. Создать класс `Vector` для работы с векторами заданной размерности. Должны быть реализованы: операции сложения и вычитания векторов, скалярное произведение векторов, вычисление Евклидовой нормы, сравнение векторов по норме. (Евклидова норма вычисляется как квадратный корень из суммы квадратов координат)

5. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям. Программа должна содержать меню, позволяющие осуществить проверку всех методов класса.

6. Описать класс «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. Программа должна содержать меню, позволяющие осуществить проверку всех методов класса.

7. Описать класс, реализующий стек. Написать программу, использующую этот класс для отыскания прохода по лабиринту.

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.

8. Описать класс “предметный указатель”. Каждый компонент указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, от одного до десяти. Предусмотреть возможность формирования указателя с клавиатуры и из файла, вывода указателя, вывода номеров страниц для заданного слова, удаления элемента из указателя.

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

9. Описать класс “множество”, позволяющий выполнять основные операции — добавление и удаление элемента, пересечение, объединение и разность множеств.

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

10. Описать класс, реализующий тип данных “вещественная матрица” и работу с ними. Класс должен реализовывать следующие операции над матрицами:

сложение, вычитание, умножение, деление (+, −, *, /) (умножение и деление, как на другую матрицу, так и на число);

комбинированные операции присваивания (+=, -=, *=, /=);

операции сравнения на равенство/неравенство;
операции вычисления обратной и транспонированной матрицы, операцию возведения в степень;
методы вычисления детерминанта и нормы;
методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная, симметрическая, верхняя треугольная, нижняя треугольная);
операции ввода/вывода в стандартные потоки.

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

11. Описать класс “студенческая группа”. Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям.

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

12. Написать класс для эффективной работы со строками, позволяющий форматировать и сравнивать строки, хранить в строках числовые значения и извлекать их. Для этого необходимо реализовать:

перегруженные операции присваивания и конкатенации;
операции сравнения и приведения типов;
преобразование в число любого типа;
форматный вывод строки.

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

13. Создать класс Payment (зарплата). В классе должны быть представлены поля: фамилия-имя-отчество, оклад, количество отработанных дней в месяце,

количество рабочих дней в месяце, начисленная и удержанная сумма. Реализовать методы: вычисления начисленной суммы, вычисление удержанной суммы, вычисление суммы, выдаваемой на руки, вычисление стажа. Стаж вычисляется как полное количество лет, прошедших от года поступления на работу, до текущего года. Начисления представляют собой сумму, начисленную за отработанные дни, и надбавки, т.е. доли от первой суммы. Удержания представляют собой отчисления в пенсионный фонд (1% от начисленной суммы) и подоходный налог составляет 13% от начисленной суммы без отчисления в пенсионный фонд.

14. Используя класс `Bill` (плательщик) реализовать класс `ListPayer`. Класс содержит список плательщиков за телефонные услуги, дату создания списка, номер списка. Поле одного элемента списка – это: плательщик класс `Bill`, признак оплаты, дата платежа, сумма платежа. Реализовать методы добавления плательщика в список и удаления из него; метод поиска плательщика по номеру телефона и по фамилии, по дате платежа. Метод вычисления полной стоимости платежей всего списка. Реализовать операцию объединения и операцию пересечения списков. Реализовать операцию генерации конкретного объекта `Group` (группа), содержащего список плательщиков, из объектов типа `ListPayer`. Должна быть возможность выбирать группу плательщиков по признаку оплаты, по атрибутам, по дате платежа, по номеру телефона.

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

16. Построить модель изображения тел звездного неба в двумерной проекции. Очевидно, что основой всякого изображения является положение (позиция базовый класс `Position`) отдельного элемента на экране, описываемая координатами X и Y . Звезда (класс `Star`) может быть видимой

или нет, ей можно задать цвет, ее можно переместить. Планета (класс Planet) имеет предком Star, обладая всеми его свойствами, кроме того, быть «больше» по размеру(точнее видимому размеру). Планеты появляются, исчезают и перемещаются. Верхний объект, закрывая нижний, не портит его.

17. Создать абстрактный базовый класс Figure с виртуальными методами вычисления площади и периметра. Создать производные классы: Rectangle (прямоугольник), Circle (круг), Trapezium (трапеция) со своими функциями площади и периметра. Самостоятельно определить, какие поля необходимы, какие из них можно задать в базовом классе, а какие – в производных. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Реализовать задание, разделив каждый класс на интерфейс и реализацию.

18. Создать абстрактный базовый класс Container с виртуальными методами sort() и поэлементной обработки контейнера foreach(). Разработать производные классы Bubble (пузырек) и Choice (выбор). В первом классе сортировка реализуется методом пузырька, а поэлементная обработка состоит в извлечении квадратного корня. Во втором классе сортировка реализуется методом выбора, а поэлементная обработка – вычисление логарифма. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Реализовать задание, разделив каждый класс на интерфейс и реализацию.

19. Создать класс ModeWindow для работы с моделями экранных окон. В качестве полей задаются: заголовок окна, координаты левого верхнего угла, размер по горизонтали, размер по вертикали, цвет окна, состояние «видимое/невидимое», состояние «с рамкой/без рамки». Координаты и размеры указываются в целых числах. Реализовать операции: передвижение окна по горизонтали, по вертикали, изменение высоты и/или ширины окна изменение цвета; изменение состояния опрос состояния. Операция передвижения и изменения размера должны осуществлять проверку на

пересечение границ экрана. Функция вывода на экран должна индуцировать состояние полей объекта. Реализовать класс, как класс-наследник с перегрузкой операций и конструкторами. В качестве базового класса реализовать класс Object со счетчиком объектов. Реализовать задание, разделив каждый класс на интерфейс и реализацию.

20.*Создать класс Triad (тройка чисел); определить методы увеличения полей на 1. Определить класс-наследник Time с полями: час, минута, секунда для работы со временем в формате «час: минута: секунда». Класс должен включать в себя не менее четырех функций инициализации: числами, строкой (например «23:59:59»), секундами и временем. Обязательными операциями являются: вычисление разницы между двумя моментами времени в секундах, сложение времени и заданного количества секунд, сравнение моментов времени, перевод в секунды, минуты. Переопределить методы увеличения полей на 1 и определить методы увеличения на n секунд и минут. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Реализовать задание, разделив каждый класс на интерфейс и реализацию.

21. Создать класс ListPayment (зарплата). В классе содержится список сотрудников, для которых рассчитывается заработная плата. Сотрудник представлен классом Person с полями :табельный номер, номер отдела, фамилия- имя – отчество, оклад, дата поступления на работу, процент надбавки, подоходный налог , количество отработанных дней в месяце, количество рабочих дней в месяце, начислено, удержано. Реализовать методы: вычисления начисленной суммы, вычисление удержанной суммы, вычисление суммы, выдаваемой на руки, вычисление стажа. Стаж вычисляется как полное количество лет, прошедших от года поступления на работу, до текущего года. Начисления представляют собой сумму, начисленную за отработанные дни , и надбавки, т.е. доли от первой суммы. Удержания представляют собой отчисления в пенсионный фонд (1% от начисленной суммы) и подоходный налог составляет 13% от начисленной

суммы без отчисления в пенсионный фонд. Реализовать методы добавления сотрудника в список и удаления из него; методы объединения списков; методы поиска по полям класса Person. Реализовать методы вычисления полных сумм по всему списку: начислено, удержано, на руки, подоходный налог, пенсионный фонд. Организовать операцию генерации объекта Group(группа), содержащего список сотрудников с одинаковым стажем работы, из объектов типа ListPayment.

Критерии оценки:

86-100 баллов - выставляется студенту, если вопрос полностью раскрыт без ошибок и даны ответы на все дополнительные вопросы

76-85 баллов - выставляется студенту, если вопрос полностью раскрыт без ошибок и даны ответы не на все дополнительные вопросы

61-75 баллов - выставляется студенту, если вопрос не полностью раскрыт без ошибок и даны ответы на все дополнительные вопросы

50-60 баллов - выставляется студенту, если вопрос не полностью раскрыт без ошибок и даны ответы не на все дополнительные вопросы