



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

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

«СОГЛАСОВАНО»
Руководитель ОП
Фундаментальная и прикладная лингвистика

«УТВЕРЖДАЮ»
Заведующий кафедрой прикладной математики, механики,
управления и программного обеспечения

Спицына Н.А.

«11» июля 2019 г.

(подпись)

Артемьева И.Л.

«11» июля 2019 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Разработка лингвистических систем

Направление подготовки 45.03.03 «Фундаментальная и прикладная лингвистика»

Форма подготовки очная

курс 4 семестр 7,8

лекции 30 час.

практические занятия 0 час.

лабораторные работы 30 час.

в том числе с использованием МАО лек 0 - / пр. 0 / лаб. 30 час

всего часов аудиторной нагрузки – 60 час.

в том числе с использованием МАО 30 -час

самостоятельная работа 84 час.

контрольные работы (количество)

курсовая работа / курсовой проект _____ семестр

зачет семестр

экзамен 7,8 семестр

Рабочая программа составлена в соответствии с требованиями образовательного стандарта, самостоятельно установленного ДВФУ, утвержденного приказом ректора ДВФУ №12-13-235 от 18.02.2016 г.

Рабочая программа обсуждена на заседании кафедры прикладной математики, механики, управления и программного обеспечения, протокол № 11 от «10» июля 2019 г.

Заведующая кафедрой прикладной математики, механики, управления и программного обеспечения Артемьева И.Л., д.т.н., профессор

Составители: заведующая кафедрой прикладной математики, механики, управления и программного обеспечения Артемьева И.Л., д.т.н., профессор
доцент кафедры прикладной математики, механики, управления и программного обеспечения Прудникова Л.И., к. ф.-м.н., доцент

АННОТАЦИЯ

Рабочая программа дисциплины "Разработка лингвистических систем" разработана для студентов 4 курса, обучающихся по направлению 45.03.03 «Фундаментальная прикладная лингвистика», все профили.

Трудоемкость дисциплины 4 зачетных единицы (144 часов). Дисциплина реализуется в 7 и 8 семестрах. В 7 семестре дисциплина содержит 18 часов лекций, 0 часов практических занятий, 18 часов лабораторных работ, из них 18 часов лабораторных работ в интерактивной форме. На самостоятельную работу студентам отводится 36 часов, из них 27 на подготовку к экзамену. В 8 семестре дисциплина содержит 12 часов лекций, 0 часов практических занятий, 12 часов лабораторных работ, из них 12 часов лабораторных работ в интерактивной форме. На самостоятельную работу студентам отводится 48 часов, из них 27 на подготовку к экзамену.

Дисциплина "Разработка лингвистических систем" базируется на дисциплинах "Основы технологии программирования", "Системный анализ при создании лингвистических систем". Учебная дисциплина «Разработка лингвистических систем» входит в раздел «Дисциплины», базовая часть. Знания, полученные при ее изучении, будут использованы при выполнении выпускной квалификационной работы бакалавра.

Цель дисциплины – развитие компетенций в области объектно-ориентированного анализа при моделировании прикладных лингвистических систем, а также изучение приемов разработки программного обеспечения с использованием методологии объектно-ориентированного программирования.

Задачи дисциплины:

1. Научить методам и инструментальным средствам объектно-ориентированного анализа и программирования, использовать инструментальные средства разработки, автономной и комплексной отладки и тестирования объектно-ориентированных программ.

2. Научить решать задачи производственной и технологической деятельности на профессиональном уровне, включая: разработку алгоритмических и программных решений; анализ предметной области и ее описание методами объектно-ориентированного анализа; применять шаблоны объектно-ориентированного программирования на языке высокого уровня (C++).

Для успешного изучения дисциплины "Разработка лингвистических систем" у обучающихся должны быть сформированы следующие предварительные компетенции: способность осуществлять поиск, хранение,

обработку и анализ информации из различных источников и баз данных; владение основными концепциями, принципами, теориями и фактами, связанными с информатикой; готовность к использованию методов разработки лингвистических систем.

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

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-1 владением основными понятиями и категориями современной лингвистики	Знает	Существующие лингвистически ориентированные продукты
	Умеет	Выбирать те лингвистически ориентированные продукты, которые требуются при создании программной системы
	Владеет	Методами использования лингвистически ориентированных продуктов при создании программ
ОПК-7 способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Знает	стандарты разработки и документирования программных средств, специфику процессов создания лингвистических комплексов программ
	Умеет	корректно ставить лингвистические задачи и создавать программные системы для их решения
	Владеет	Методами анализа приложений, проектирования программных средств по результатам анализа, оценивания полученного программного продукта
ОПК-2 владением основами математических дисциплин, необходимых для формализации лингвистических знаний и процедур анализа и синтеза лингвистических структур	Знает	Существующие методы обработки лингвистических данных.
	Умеет	Использовать существующие лингвистические системы при проектировании новых
	Владеет	методами алгоритмического моделирования при анализе и разработке лингвистических задач
ОК-5 способность использовать современные методы и технологии (в том числе информационные) в профессиональной деятельности	Знает	Методы организации тестирования программных систем
	Умеет	Проектировать требуемый набор тестов для оценки качества лингвистических приложений

	Владеет	Навыками создания пакета тестов для оценки качества лингвистических приложений
--	---------	--

Для формирования вышеуказанных компетенций в рамках дисциплины "Разработка лингвистических систем" применяются следующие методы активного/ интерактивного обучения: метод новых проектов.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Трудоемкость теоретической части курса 30 часов

Раздел.1. Базовые понятия. (5 ч.)

Тема 1. Введение (1 ч.)

Место дисциплины "Разработка лингвистических систем". Формальные средства построения модели предметной области. Объектно-ориентированная методология.

Тема 2. Объектно-ориентированный анализ(4ч.)

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 C++(13ч.)

Тема 3. Программа на языке C++ (3ч.)

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

Тема 4. Синтаксис и семантика языка C++ (8ч.)

Преобразование типов данных. Массивы и строковые данные. Поразрядные логические операции и операции сдвигов. Логические выражения. Передача параметров. Прототипы функций в C++. 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 C++.

Тема 5. Данные типа структура в языке C++ (2ч.)

Структура в языке C++. Списки и их реализация.

Раздел.3. Объектно-ориентированное проектирование и программирование (ООП) (12ч.)

Тема 6. Основные принципы(6ч.)

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

Тема 7. Шаблоны(6ч.)

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

II. СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Лабораторные работы(30ч.)

Лабораторные работы № 1. Алгоритмы.(6ч.)

Лабораторные работы № 2. Простейший тип программы.(4ч.)

Лабораторные работы № 3. Конструкторы.(8ч.)

Лабораторные работы № 4. Составные типы (4ч.)

Лабораторные работы № 5. Строки.(4ч.)

Лабораторные работы № 6. Подпрограммы (4ч.)

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине "Разработка лингвистических систем" 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-1, ОПК-7, ОПК-2, ОК-5	Знает	ПР7 конспект	Вопросы к экзамену 1-10
2.	Раздел.2. Объектное программирование на С++	ОПК-1, ОПК-7, ОПК-2, ОК-5	Знает	ПР7 конспект	Вопросы к экзамену 11-18
3.	Раздел.3. Объектно- ориентированное проектирование и программирование(О ОП)	ОПК-1, ОПК-7, ОПК-2, ОК-5	Знает	ПР7 конспект	Вопросы к экзамену 19-25
6.	Лабораторные работы №1-6	ОПК-1, ОПК-7, ОПК-2, ОК-5	умеет, владеет	ПР6 лабораторная работа	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

1. Дейл Н., Уимз Ч. Программирование на С++. Изд.: ДМК Пресс, 2007-672 с. [электронный ресурс] - Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=1219
2. Долгов, А. И. Алгоритмизация прикладных задач. Уч. пособ.- М.: Флинта, 2011. - 136 с. [электронный ресурс] - Режим доступа: <http://znanium.com/bookread.php?book=406093>

3. Объектно-ориентированное программирование в C++: лекции и упражнения [Электронный ресурс]: Учебное пособие для вузов / Ашарина И.В. - М.: Горячая линия - Телеком, 2012. - <http://www.studentlibrary.ru/book/ISBN9785991270014.html>
4. Подбельский В.В., Фомин С.С. Программирование на языке Си: Учеб. пособие для вузов. М.: Финансы и статистика, 2009. - 600 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:355876&theme=FEFU>
5. C/C++. Программирование на языке высокого уровня: учебник для вузов для магистров и бакалавров / Т. А. Павловская. СПб.: Питер, 2011.-460 с. <https://lib.dvfu.ru:8443/lib/item?id=chamo:307692&theme=FEFU>
6. Язык программирования C++. Специальное издание: пер. с англ. / Бьерн Страуструп. - М.: Вильямс, 2011. — 1135с. <https://lib.dvfu.ru:8443/lib/item?id=chamo:418460&theme=FEFU>

Дополнительная литература

1. Прудникова Л.И. Программирование на языке C/C⁺⁺: учебно-методич. пособие / Л.И. Прудникова. – Влад. : Изд.дом ДВФУ, 2012. – 144 с.
2. Ларман, Крэг Применение UML и шаблонов проектирования: введение в объектно-ориентированный анализ, проектирование и унифицированный процесс UP.- М.: Издательский дом Вильямс , 2004,- 624 с.
3. Керниган Б., Ритчи Д. Язык программирования С. Пер. с англ. — М.: Издательский дом "Вильямс", 2012. — 289 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:666721&theme=FEFU>
4. Подбельский В.В., Фомин С.С. Программирование на языке Си: Учеб. пособие для вузов. М.: Финансы и статистика, 2009. - 600 с.. <http://lib.dvfu.ru:8080/lib/item?id=chamo:355876&theme=FEFU>
5. Хабибуллин И.Ш. Программирование на языке высокого уровня C/C⁺⁺. - СПб.: БХВ-Петербург, 2006. - 499 с. <http://znanium.com/catalog.php?bookinfo=356906>
6. Липман С., Лажоие Ж. Язык программирования C++. Полное руководство. Изд.: ДМК Пресс, 2006 -1105 с. [электронный ресурс] - Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=12162
7. Забудский Е.И. "Объектно-ориентированный анализ и программирование на языке С# - Пособие к практическим занятиям". Издъ: М.: Кафедра ОИиППО ГУ-ВШЭ, 2005 - 440 с.
8. Ноткин А.М. Объектно-ориентированное программирование на языке C++. Лабораторный практикум. — Пермь: Перм. гос. техн. ун-т, 2001. - 92 с.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. <http://www.kodges.ru/5537-jazyk-s-i-obektno-orientirovannoe.html> Скляр В.А. Язык С++ и объектно-ориентированное программирование. Справочное пособие. Изд.: Мн.: 'Вышэйшая школа', 1997- 481с. [электронный ресурс]
2. <http://window.edu.ru/resource/458/24458/files/metoopk.pdf> Объектно-ориентированная методология разработки сложных систем: Учебное пособие / Т.В. Глотова. Пенза, 2001. – 49 с.
3. <http://www.razym.ru/94003-grady-buch-robert-a-maksimchuk-majkl-u-yengl.html> Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч, Р. Максимчук, М. Энгл, Б. Янг, Д. Коннален, К. Хьюстон. – М.: Издательский дом «Вильямс», 2008 – 720 с. [электронный ресурс]
4. <https://eknigi.org/programmirovanie/53044-obektno-orientirovannyj-analiz-i-proektirovanie.html> Йордон Э., Аргила К.. Объектно-ориентированный анализ и проектирование систем. Изд.: Лори, 2007 – 284с. [электронный ресурс]
5. <https://code-live.ru/tag/cpp-manual/> Портал о программировании
6. <http://cppstudio.com/cat/274/> Язык программирования С++
7. <http://www.c-cpp.ru/> Программирование на С и С++. Онлайн справочник программиста
8. <http://www.proklondike.com/books/cpp.html> Программирование
9. <http://purecodecpp.com/> Основы программирования на С++ для начинающих
10. <http://progbook.ru/c/> Книги по С++
11. <http://mexalib.com/view/2267> Компьютерная литература. Шилдт Г. Самоучитель С++ ВHV - Санкт - Петербург, 2003. - 688 стр.
12. <http://www.programmersclub.ru/main/> Клуб программистов
13. <http://info-comp.ru/> Информационный портал. Все о компьютере и программировании для начинающих

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используется следующее программное обеспечение:

1. Пакет прикладных программ Microsoft Office / Open Office.
2. Интегрированные среды разработки программ на языке C++.
3. Программное обеспечение электронного ресурса сайта ДВФУ, включая ЭБС ДВФУ.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующие информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.
2. Электронно-библиотечная система IPRbooks.
3. Доступ к электронному заказу книг в библиотеке ДВФУ, доступ к нормативным документам ДВФУ, расписанию, рассылке писем.

Лабораторные занятия проводятся в специализированном компьютерном классе.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

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.
7. Массивы в языке СИ. Строковые данные.
8. Константы литерные и арифметические.
9. Побитовые операции и моделирование работы с множеством.
10. Поразрядные логические операции и операции сдвигов. Примеры. Задача.
11. Логические выражения.

Вопросы для самоконтроля по разделу 2

1. Указатели. Операции над ними.
2. Функция в языке СИ. Передача параметров. Передача функции как параметра.
3. Структура в языке СИ. Примеры.
4. Списки, очереди, стеки, деревья и их реализация. ПОЗ.
5. Классы памяти и области видимости переменных и функций.
6. Синтаксис описания и объявления в языке СИ. Сложные объявления.
7. Прототипы функций в СИ++. Шаблоны функций. Перегрузка функций.
8. Ссылки.

Вопросы для самоконтроля по разделу 3

1. Объектно-ориентированное проектирование и программирование.
2. Основные принципы. Основные компоненты.

3. Объект и его свойства. Операции над объектами. Отношения между объектами.
4. Класс. Отношения между классами.
5. Объект и класс в СИ++. Область видимости. Наследование.
6. Шаблоны классов. Пример программы.
7. Константы. Перечислимый тип. Ввод операций в СИ++.
8. Стили программирования и методология ООП.
9. 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 DLPPProjectorPT-D2110XE, плазма LG FLATRON M4716CCBAM4716CJ. Для выполнения самостоятельной работы студенты в жилых корпусах ДВФУ обеспечены Wi-Fi.



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

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ
ОБУЧАЮЩИХСЯ**

по дисциплине «Разработка лингвистических систем»

Направление подготовки 45.03.03 «Фундаментальная прикладная лингвистика»

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.	2-9-я недели	Выбор темы ИПС №1. Разработка алгоритма. Разработка ПО для ИПС.	18 часов	проект
2.	10-18-я недели	Выбор темы ИПС № 2. Разработка алгоритма. Разработка ПО для ИПС.	18 часов	проект
3.	19-24-я недели	Выбор темы ИПС №3. Разработка алгоритма. Разработка ПО для ИПС.	24 часа	проект
4.	25-30-я недели	Выбор темы ИПС №4. Разработка алгоритма. Разработка ПО для ИПС.	34 часа	проект

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 ЕСТЕСТВЕННЫХ НАУК

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Разработка лингвистических систем»

Направление подготовки 45.03.03 «Фундаментальная прикладная лингвистика»

Форма подготовки (очная)

Владивосток
2016

Паспорт ФОС

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-1 владением основными понятиями и категориями современной лингвистики	Знает	Существующие лингвистически ориентированные продукты
	Умеет	Выбирать те лингвистически ориентированные продукты, которые требуются при создании программной системы
	Владеет	Методами использования лингвистически ориентированных продуктов при создании программ
ОПК-7 способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Знает	стандарты разработки и документирования программных средств, специфику процессов создания лингвистических комплексов программ
	Умеет	корректно ставить лингвистические задачи и создавать программные системы для их решения
	Владеет	Методами анализа приложений, проектирования программных средств по результатам анализа, оценивания полученного программного продукта
ОПК-2 владением основами математических дисциплин, необходимых для формализации лингвистических знаний и процедур анализа и синтеза лингвистических структур	Знает	Существующие методы обработки лингвистических данных.
	Умеет	Использовать существующие лингвистические системы при проектировании новых
	Владеет	методами алгоритмического моделирования при анализе и разработке лингвистических задач
ОК-5 способность использовать современные методы и технологии (в том числе информационные) в профессиональной деятельности	Знает	Методы организации тестирования программных систем
	Умеет	Проектировать требуемый набор тестов для оценки качества лингвистических приложений
	Владеет	Навыками создания пакета тестов для оценки качества лингвистических приложений

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1	Раздел.1. Базовые понятия	ОПК-1, ОПК-7, ОПК-2, ОК-5	Знает	ПР7 конспект	Вопросы к экзамену 1-10
2.	Раздел.2. Объектное программирование на C++	ОПК-1, ОПК-7, ОПК-2, ОК-5	Знает	ПР7 конспект	Вопросы к экзамену 11-18
3.	Раздел.3. Объектно- ориентированное проектирование и программирование(О ОП)	ОПК-1, ОПК-7, ОПК-2, ОК-5	Знает	ПР7 конспект	Вопросы к экзамену 19-25
6.	Лабораторные работы №1-6	ОПК-1, ОПК-7, ОПК-2, ОК-5	умеет, владеет	ПР6 лабораторная работа	Отчет

Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
ОПК-1 владением основными понятиями и категориями современной лингвистики	знает (пороговый уровень)	Существующие лингвистически ориентированные продукты	Знание существующих лингвистических систем	Способность раскрыть суть технологии разработки программ дать ответы на вопросы
	умеет (продвинутый)	Выбирать те лингвистически ориентированные продукты, которые требуются при создании программной системы	Умение выбирать лингвистические программные системы	Способность обосновать выбор

	владеет (высокий)	Методами использован ия лингвистически ориентированных продуктов при создании программ	Владение набором средств, предоставляемых лингвистическими системами	Способность работать с различными лингвистическими системами
ОПК-7 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	знает (пороговый уровень)	стандарты разработки и документирования программных средств, специфику процессов создания лингвистических комплексов программ	Знание стандартов разработки и документирования программных средств, специфики процессов создания больших комплексов программ.	Способность дать ответы на вопросы
	умеет (продвинутый)	корректно ставить лингвистические задачи и создавать программные системы для их решения	Умение корректно ставить лингвистические задачи и организовывать их решение в рамках небольших коллективов	Способность работать с различными стандартами разработки и документирования программных средств
	владеет (высокий)	Методами анализа приложений, проектирования программных средств по результатам анализа, оценивания полученного программного продукта	Владение методами разработки программной системы с использованием языкового процессора объектно-ориентированного языка	Наличие созданных проектов

ОПК-2 владением основами математичес ких дисциплин, необходимы х для формализац ии лингвистиче ских знаний и процедур анализа и синтеза лингвистиче ских структур	знает (пороговый уровень)	Существую щие методы обработки лингвистиче ских данных.	Знание основных методов обработки лингвистическ их данных.	Способность дать ответы на вопросы
	умеет (продвинутый)	Используй вать существую щие лингвистиче ские системы при проектирова нии новых	Умение использовать существующие лингвистическ ие системы	Наличие созданных проектов
	владеет (высокий)	методами алгоритмиче ского моделирова ния при анализе и разработке лингвистиче ских задач	Владение методами разработки объектно- ориентированн ых программ в некоторой среде программирован ия	Наличие созданных программ
ОК-5 способность использоват ь современны е методы и технологии (в том числе информацио нные) в профессиона льной деятельност и	знает (пороговый уровень)	Методы организации тестировани я программны х систем	Знание методов проектировани я тестовых ситуаций и тестов	Способность дать ответы на вопросы
	умеет (продвинутый)	Проектиров ать требуемый набор тестов для оценки качества лингвистиче ских приложений	Умение определить тестовые ситуации и необходимый набор тестов для проверки качества создаваемых программ	Наличие созданных проектов

	владеет (высокий)	Навыками создания пакета тестов для оценки качества лингвистиче ских приложений	Умение записать тестовые данные средствами объектно- ориентированно го языка	Наличие созданных тестов
--	----------------------	---	---	--------------------------

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 (рейтинговая оценка)	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 «Разработка лингвистических систем»

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)

19. Объектно-ориентированное проектирование и программирование.
20. Основные принципы. Основные компоненты.
21. Объект и его свойства. Операции над объектами. Отношения между объектами.
22. Класс. Отношения между классами.
23. Объект и класс в СИ++. Область видимости. Наследование.
24. Шаблоны классов. Пример программы.
25. Константы. Перечислимый тип. Ввод операций в СИ++.
26. Стили программирования и методология ООП.
27. 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 А, В, С так, чтобы внутри треугольника АВС 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 А размером 20×20 . Считая ее составленной из 100 квадратов размером 2×2 и переставляя эти квадраты, преобразовать А так,

чтобы в результирующей матрице для всяких двух квадратов В и С выполнялось следующее условие; если сумма элементов В меньше суммы элементов С, то В лежит либо выше, либо левее (когда В и С на одной горизонтали) квадрата С.

24. На плоскости задано множество окружностей. Две окружности А и В назовем связанными, если они пересекаются, либо существует третья окружность С заданного множества, связанная с А и В. 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*.

1. 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*.

2. 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.

3. 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.
4. 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.
5. В заданном предложении указать слово, в котором доля гласных (А, Е, У, I, О) максимальна.

6. Переставить и распечатать слова заданного предложения в соответствии с ростом доли согласных (В, С, D, F, G, H, К, L, M, N, P, Q, R, S, T, V, W, X, Z) в этих словах.
7. 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.

2. 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.
3. 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...

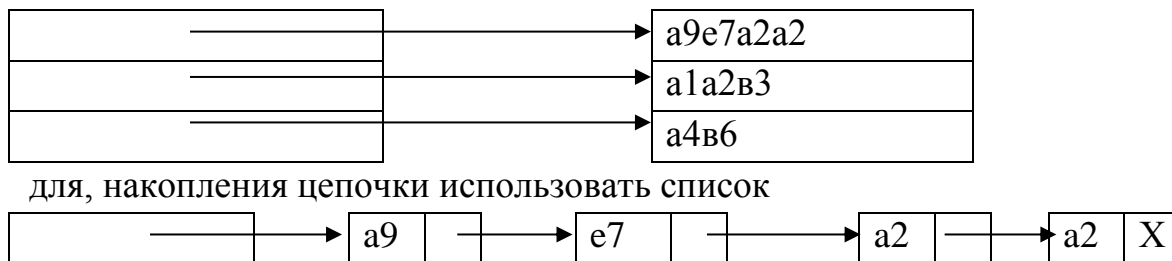


Рис.52.

Для ускорения поиска иметь вспомогательный указатель на последний

элемент списка.

4. 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.

5. Во входном потоке встречаются в произвольном порядке подцепочки $alal+alalal * a1 alalal \dots$

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

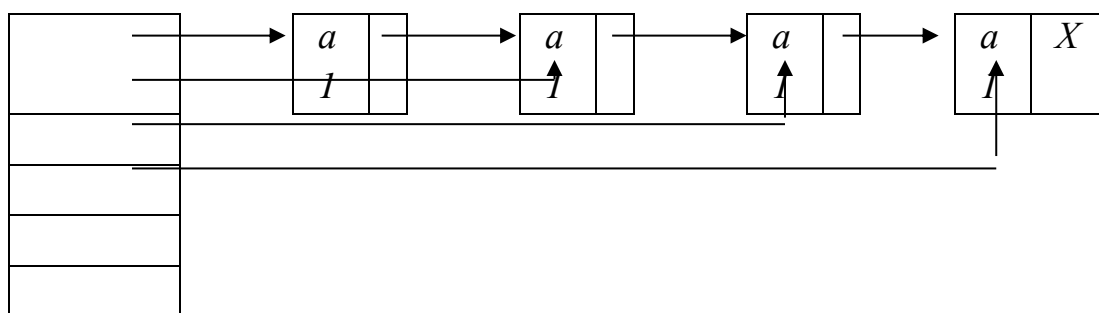


Рис.53.

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

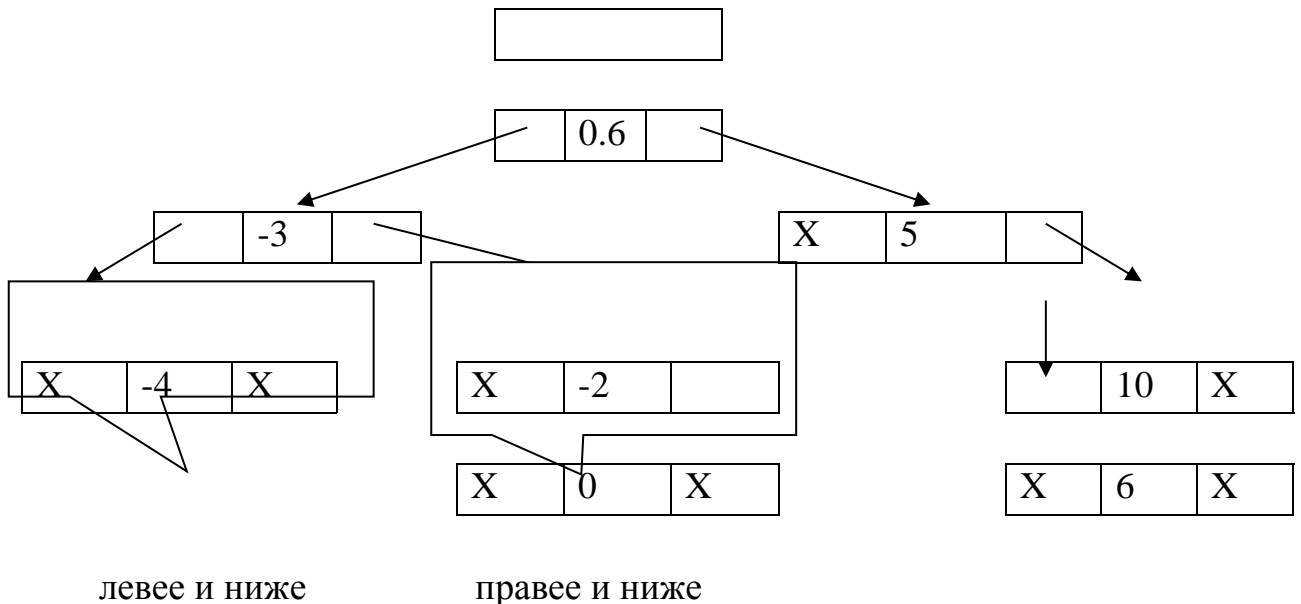
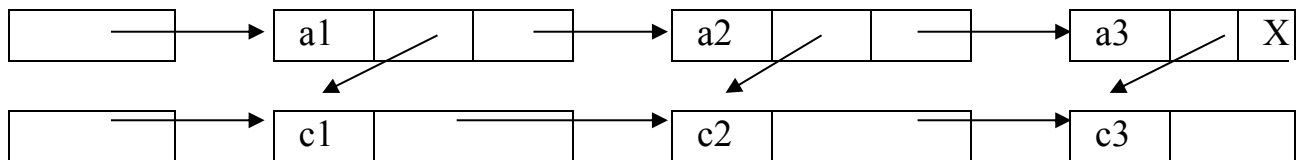


Рис.55.

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