Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники
Кафедра автоматизированных систем управления
Кафедра автоматики

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н. Рева И. Л.
“___ ”_____ _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Программирование

Образовательная программа: 09.03.01 Информатика и вычислительная техника, профиль: Программное обеспечение компьютерных систем и сетей

Курс: 1 2, семестры : 2 3

Факультет автоматики и вычислительной техники, заочная форма обучения

		Семестр	
№	**Вид деятельности**	**2**	**3**
1	Всего зачетных единиц (кредитов)	0	6
2	Всего часов	0	216
3	Всего занятий в контактной форме, час.	2	40
4	Лекции, час.	2	4
5	Практические занятия, час.	0	2
6	Лабораторные занятия, час.	0	12
7	из них в активной и интерактивной форме, час.	0	2
8	Аттестация, час.	0	2
9	Консультации, час.		20
10	Самостоятельная работа, час.	0	174
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)		КР
12	Вид аттестации		Э

Новосибирск 2017

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.03.01 Информатика и вычислительная техника

ФГОС введен в действие приказом №5 от 12.01.2016 г. , дата утверждения: 09.02.2016 г.

Место дисциплины в структуре учебного плана: Блок 1, базовая

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.03.01 Информатика и вычислительная техника

Рабочая программа обсуждена на заседании кафедры

ВТ, протокол заседания кафедры №6 от 20.06.2017
АСУ, протокол заседания кафедры №7 от 20.06.2017
АВТ, протокол заседания кафедры №10/1 от 20.06.2017

Утверждена на совете факультета автоматики и вычислительной техники, протокол № 6 от 21.06.2017

Программу разработал:

доцент, к.ф-м.н. Гейнц О. М.

Заведующий кафедрой:

доцент, к.т.н. Якименко А. А.
профессор, д.т.н. Гриф М. Г.
доцент, д.т.н. Жмудь В. А.

Ответственный за образовательную программу:

доцент Романов Е. Л.

1. Внешние требования

Таблица 1.1

***Компетенция ФГОС*: ОПК.5 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; *в части следующих результатов обучения*:**
з4. знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах
у12. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе
у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств
***Компетенция ФГОС*: ПК.3 способность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности; *в части следующих результатов обучения*:**
у7. уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности
***Компетенция НГТУ*: ПК.9.В/ПК готовность к разработке моделей компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"; *в части следующих результатов обучения*:**
з2. знать основы объектно-ориентированного подхода к программированию

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
Программирование	
ПК.9.В/ПК.з2 знать основы объектно-ориентированного подхода к программированию	
1.знать основы объектно-ориентированного подхода к программированию	Лекции; Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.3.у7 уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	
2.уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	Лекции; Практические занятия; Лабораторные работы; Самостоятельная работа
ОПК.5.з4 знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах	
3.знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах	Лекции; Практические занятия; Лабораторные работы; Самостоятельная работа
ОПК.5.у12 уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	
4.уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	Лекции; Практические занятия; Лабораторные работы; Самостоятельная работа
ОПК.5.у5 уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	
5.уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	Лекции; Практические занятия; Лабораторные работы; Самостоятельная работа

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения
Семестр: 2			
Дидактическая единица: "Эпизодическое" объектно-ориентированное программирование			
1. Объекты и классы Технологическое определение класса. Синтаксическое определение объекта и класса - структура со встроенными функциями. Контекст класса Классы и порождение объектов	0	2	1, 3, 5
Семестр: 3			
Дидактическая единица: Модульное проектирование и ООП.			
2. Свойства объекта: закрытость, независимость, универсальность, целостность. Целостность объекта. Конструктор. Деструктор Безымянные объекты. Внутреннее программирование Внешнее программирование	0	2	1, 3, 4, 5
Дидактическая единица: Переопределение операций			
3. Переопределение операций внутри класса Переопределение операции присваивания. Переопределение операции приведения типа. Переопределение операций () и []	0	2	1, 2, 3, 4, 5

Таблица 3.2

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 3				
Дидактическая единица: Модульное проектирование и ООП.				
1. Разработка классов, создание конструкторов и деструкторов. Ис-пользование статических членов класса.	1	4	1, 2, 3, 4, 5	Выполнение задания по вариантам
Дидактическая единица: Программирование методов и принцип модульности				
3. Разработка классов, создание конструкторов и деструкторов. Ис-пользование статических членов класса.	0	4	1, 2, 3, 4, 5	Выполнение задания по вариантам
Дидактическая единица: Переопределение операций				
2. С использованием синтаксиса переопределения операций разработать стандартную арифметику объектов. Использовать задания Л.р. №1	1	4	1, 2, 3, 4, 5	Выполнение задания по вариантам

Таблица 3.3

Темы практических занятий	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность

Семестр: 3				
Дидактическая единица: Переопределение операций				
1. Переопределение операций внутри класса Переопределение операции присваивания.	0	2	1, 2, 3, 4, 5	Решение задач по теме

Таблица 3.4

Темы для самостоятельного изучения	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 2				
Дидактическая единица: "Эпизодическое" объектно-ориентированное программирование				
4. Класс. Класс - граница ответственности транслятора и программы. Присваивание, передачу по значению в функцию и возвращение его в виде значения-результата.	0	2	1, 2, 3, 4, 5	Изучение теоретического материала
Семестр: 3				
Дидактическая единица: Модульное проектирование и ООП.				
1. Внутреннее программирование Внешнее программирование Объявление дружественной функции Эпизодическое объектно-ориентированное программирование. Тотальное ООП и программирование "от класса к классу". Особенности модульного проектирования в технологии ООП	0	10	1, 2, 3, 4	Самостоятельное изучение теоретического материала
Дидактическая единица: Программирование методов и принцип модульности				
2. Способы передачи параметров в методы Конструктор копирования Конструктор копирования для случая разделения данных. Конвейер ссылок и конвейер значений Статические элементы класса	0	10	1, 2, 3, 4, 5	Самостоятельное изучение теоретического материала
Дидактическая единица: Переопределение операций				
3. Переопределение операций new и delete. Переопределение операции вне класса	0	10	1, 2, 3, 4, 5	Изучение теоретического материала

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 2				
1	Самостоятельное изучение теоретического материала	1, 2, 3, 4, 5	2	0
Студент изучает темы, приведенные в таблице 3.4 : Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана. Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана.				
Семестр: 3				
1	Курсовая работа	1, 2, 3, 4, 5	40	5
Выполнение индивидуальных заданий по вариантам, подробная информация приведена в приложениях №1 №3 : Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана. Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана.				
2	Подготовка к занятиям	1, 2, 3, 4, 5	30	10
Для лабораторных занятий - отладка программ, оформление отчетов, выполнение тестовых заданий к защите : Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана. Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана.				
3	Дополнительная учебная деятельность	1, 2, 3, 4, 5	38	0
Самостоятельное изучение определенных тем теоретического курса, не вошедших в лекции: Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана. Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана.				
4	Подготовка к аттестации	1, 2, 3, 4, 5	36	5
: Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана. Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана.				
5	Самостоятельное изучение теоретического материала	1, 2, 3, 4, 5	30	0

Студент изучает темы, приведенные в таблице 3.4 : Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана. Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана.

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	e-mail; Портал НГТУ; Среда электронного обучения НГТУ
Консультирование	e-mail; Среда электронного обучения НГТУ
Контроль	e-mail; Среда электронного обучения НГТУ
Размещение учебных материалов	e-mail; Среда электронного обучения НГТУ

Таблица 5.2

Активные и интерактивные формы проведения занятий

№	Наименование активных форм	Коды формируемых компетенций
1	Деловая игра	ОПК.5; ПК.3; ПК.9.В/ПК
Формируемые умения: з2. знать основы объектно-ориентированного подхода к программированию; з4. знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах; у12. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе; у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств; у7. уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности		
Краткое описание применения: Разработка алгоритмов и программ командами по 3-4 человека.		
Подробная информация об использовании технологии приводится в "Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана."		
2	Метод проектов	ПК.3;
Формируемые умения: у7. уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности		
Краткое описание применения: Представление и защита выполненных самостоятельно работ по заранее определенной теме		
Подробная информация об использовании технологии приводится в "Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана."		

3	Портфолио	ОПК.5; ПК.9.В/ПК
Формируемые умения: з2. знать основы объектно-ориентированного подхода к программированию; у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств		
Краткое описание применения: Представление и защита выполненных самостоятельно работ по заранее определенной теме		
Подробная информация об использовании технологии приводится в "Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана."		
4	Проблемный метод	ОПК.5;
Формируемые умения: з4. знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах; у12. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе		
Краткое описание применения: Представление и защита выполненных самостоятельно работ по заранее определенной теме		
Подробная информация об использовании технологии приводится в "Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана."		

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставлять оценки по традиционной шкале и 15-уровневой ECTS. Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл
Семестр: 3		
Лабораторная:	10	20
Контролирующие материалы (список вопросов) приводятся в "Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана."		
Курсовая работа: Итого	0	40
Контролирующие материалы (список вопросов) приводятся в "Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана."		
Экзамен:	0	40
Контролирующие материалы (список вопросов) приводятся в "Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана."		

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	Результаты обучения	Формы контроля		
		Защита Л/Р	Защита КП/КР	Экзамен
ОПК.5	з4. знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах	+	+	+
	у12. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	+	+	+
	у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	+	+	+
ПК.3	у7. уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	+	+	+
	ПК.9.В/ПК з2. знать основы объектно-ориентированного подхода к программированию	+	+	+

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Романов Е. Л. Си/Си ++. От дилетанта до профессионала [Электронный ресурс] : электронное учебное пособие : для 1-2 курсов направления 230100 "Информатика и вычислительная техника / Романов Е. Л. - Новосибирск, 2010. - 1 электрон. опт. диск (CD-ROM). - Загл. с этикетки диска.- Рег. свидетельство №18891. - Режим доступа:http://elibrary.nstu.ru/source?bib_id=vtls000134024

2. Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана.

3. Подбельский В. В. Язык Си++ : [учебное пособие для вузов по направлениям "Прикладная математика" и "Вычислительные машины, комплексы, системы и сети"] / В. В. Подбельский. - М., 2007. - 559 с. : ил., табл.

4. Хабибуллин И. . Программирование на языке высокого уровня С/С++ : учебное пособие для вузов по направлению 654600 "Информатика и вычислительная техника" / И. Ш. Хабибуллин. - СПб, 2006. - 485 с. : ил.

Дополнительная литература

1. Романов Е. Л. Язык программирования СИ и организация данных : Конспект лекций по дисц. "Информатика" для 11 курса фак. автомат. и вычисл. техники дн. и заоч. форм обуч. (спец. 2201,2204) / Новосиб. гос. техн. ун-т. - Новосибирск, 1996. - 100 с.

2. Романов Е. Л. Язык Си++ в задачах, вопросах и ответах : [учебное пособие] / Е. Л. Романов. - Новосибирск, 2003. - 426, [1] с. : ил.

3. Язык Си и введение в системное программирование в MS DOS : Метод. указания к лаб. работам по дисциплине "Программироваание", разд. "Язык Си и введ. в систем. программирование" для 2 курса АВТФ всех спец. / Новосиб. гос. техн. ун-т; [Сост. : Романов Е. Л.]. - Новосибирск, 1993. - 64 с.

4. Подбельский В. В. Практикум по программированию на языке Си : учебное пособие для вузов по направлениям "Прикладная математика и информатика", "Информатика и вычислительная техника" и специальности "Прикладная математика и информатика" / В. В. Подбельский. - М., 2004. - 574, [1] с. + 1 электрон. опт. диск (CD-ROM).

5. Климова Л. М. СИ++. Практическое программирование. Решение типовых задач : учебное пособие / Л. М. Климова. - М., 2001. - 587 с. : ил.

Интернет-ресурсы

1. Юн С. Г. Программирование [Электронный ресурс] : электрон. учеб.-метод. комплекс / С. Г. Юн, Е. Л. Романов; Новосиб. гос. техн. ун-т. – Новосибирск, 2012. – Режим доступа: http://dispace.edu.nstu.ru/didesk/course/show/2271. – Загл. с экрана.

2. ЭБС НГТУ : http://elibrary.nstu.ru/

3. ЭБС «Издательство Лань» : https://e.lanbook.com/

4. ЭБС IPRbooks : http://www.iprbookshop.ru/

5. ЭБС "Znanium.com" : http://znanium.com/

6. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Романов Е. Л. Си/Си++. От дилетанта до профессионала [Электронный ресурс] : [электронный учебно-методический комплекс] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000163868. - Загл. с экрана.

8.2 Специализированное программное обеспечение

1 Visual C++
2 Office

9. Материально-техническое обеспечение

Презентационное оборудование

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Лекции

Компьютерный класс

№	Наименование	Назначение
1	Компьютерный класс (Компьютеры объединены в локальную сеть с выходом в Internet)	Л.Р., защита К.Р

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра автоматизированных систем управления
Кафедра автоматики
Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева
“___ ”_____ _____ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Программирование

Образовательная программа: 09.03.01 Информатика и вычислительная техника, профиль: Программное обеспечение компьютерных систем и сетей

Новосибирск 2017

1. **Обобщенная структура фонда оценочных средств учебной дисциплины**

Обобщенная структура фонда оценочных средств по **дисциплине** Программирование приведена в Таблице.

Таблица

Формируемые компетенции	**Показатели сформированности компетенций (знания, умения, навыки)**	**Темы**	**Этапы оценки компетенций**	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	**Промежуточная аттестация (экзамен, зачет)**
ОПК.5 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	з4. знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах	Внутреннее программирование Внешнее программирование Объявление дружественной функции Эпизодическое объектно-ориентированное программирование. Тотальное ООП и программирование "от класса к классу". Особенности модульного проектирования в технологии ООП Класс. Класс - граница ответственности транслятора и программы. Присваивание, передачу по значению в функцию и возвращение его в виде значения-результата. Объекты и классы Технологическое определение класса. Синтаксическое определение объекта и класса - структура со встроенными функциями. Контекст класса Классы и порождение объектов Переопределение операций внутри класса Переопределение операции присваивания. Переопределение операции приведения типа. Переопределение операций () и [] Разработка классов, создание конструкторов и деструкторов. Ис-пользование статических членов класса. С использованием синтаксиса переопределения операций разработать стандартную арифметику объектов. Использовать задания Л.р. №1 Свойства объекта: закрытость, независимость, универсальность, целостность. Целостность объекта. Конструктор. Деструктор Безымянные объекты. Внутреннее программирование Внешнее программирование	Курсовая работа. Отчеты по лабораторным работам №1,№2,№3	Экзамен, вопросы 1-35
ОПК.5	у5. уметь применять основные методы, способы и средства получения, хранения и	Объекты и классы Технологическое определение класса. Синтаксическое определение объекта и класса - структура со встроенными	Курсовая работа. Отчет по лабораторной работе №1.	Экзамен, вопросы 20-26

	переработки информации с помощью компьютеров и компьютерных средств	функциями. Контекст класса Классы и порождение объектов Переопределение операций new и delete. Переопределение операции вне класса Переопределение операций внутри класса Переопределение операции присваивания. Разработка классов, создание конструкторов и деструкторов. Ис-пользование статических членов класса. С использованием синтаксиса переопределения операций разработать стандартную арифметику объектов. Использовать задания Л.р. №1		
ОПК.5	у12. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	Внутреннее программирование Внешнее программирование Объявление дружественной функции Эпизодическое объектно-ориентированное программирование. Тотальное ООП и программирование "от класса к классу". Особенности модульного проектирования в технологии ООП Класс. Класс - граница ответственности транслятора и программы. Присваивание, передачу по значению в функцию и возвращение его в виде значения-результата. Переопределение операций new и delete. Переопределение операции вне класса Переопределение операций внутри класса Переопределение операции присваивания. Переопределение операций внутри класса Переопределение операции присваивания. Переопределение операции приведения типа. Переопределение операций () и [] Разработка классов, создание конструкторов и деструкторов. Ис-пользование статических членов класса. С использованием синтаксиса переопределения операций разработать стандартную арифметику объектов. Использовать задания Л.р. №1 Свойства объекта: закрытость, независимость, универсальность, целостность. Целостность объекта. Конструктор. Деструктор Безымянные объекты. Внутреннее программирование Внешнее программирование Способы передачи параметров в методы	Курсовая работа. Отчет по лабораторной работе №2	Экзамен, вопросы 20-30

		Конструктор копирования Конструктор копирования для случая разделения данных. Конвейер ссылок и конвейер значений Статические элементы класса		
ПК.3/НИ готовность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	у7. уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	Внутреннее программирование Внешнее программирование Объявление дружественной функции Эпизодическое объектно-ориентированное программирование. Тотальное ООП и программирование "от класса к классу". Особенности модульного проектирования в технологии ООП Класс. Класс - граница ответственности транслятора и программы. Присваивание, передачу по значению в функцию и возвращение его в виде значения-результата. Переопределение операций new и delete. Переопределение операции вне класса Переопределение операций внутри класса Переопределение операции присваивания. Разработка классов, создание конструкторов и деструкторов. Ис-пользование статических членов класса. С использованием синтаксиса переопределения операций разработать стандартную арифметику объектов. Использовать задания Л.р. №1 Способы передачи параметров в методы Конструктор копирования Конструктор копирования для случая разделения данных. Конвейер ссылок и конвейер значений Статические элементы класса	Курсовая работа. Отчет по лабораторной работе №3	Экзамен, вопросы 26 -35
ПК.9.В/ПК готовность к разработке моделей компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"	з2. знать основы объектно-ориентированного подхода к программированию	Внутреннее программирование Внешнее программирование Объявление дружественной функции Эпизодическое объектно-ориентированное программирование. Тотальное ООП и программирование "от класса к классу". Особенности модульного проектирования в технологии ООП Объекты и классы Технологическое определение класса. Синтаксическое определение объекта и класса - структура со встроенными функциями. Контекст класса Классы и порождение объектов Переопределение операций внутри класса Переопределение операции присваивания.	Курсовая работа Отчет по лабораторной работе №1-№3	Экзамен, вопросы 20-35

		Переопределение операции приведения типа. Переопределение операций () и [] Разработка классов, создание конструкторов и деструкторов. Ис-пользование статических членов класса. С использованием синтаксиса переопределения операций разработать стандартную арифметику объектов. Использовать задания Л.р. №1 Свойства объекта: закрытость, независимость, универсальность, целостность. Целостность объекта. Конструктор. Деструктор Безымянные объекты. Внутреннее программирование Внешнее программирование		

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по **дисциплине** проводится в 3 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОПК.5, ПК.3/НИ, ПК.9.В/ПК.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 3 семестре обязательным этапом текущей аттестации является курсовая работа. Требования к выполнению курсовой работы, состав и правила оценки сформулированы в паспорте курсовой работы.

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.5, ПК.3/НИ, ПК.9.В/ПК, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения

учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра автоматизированных систем управления
Кафедра автоматики
Кафедра вычислительной техники

Паспорт экзамена

по дисциплине «Программирование», 3 семестр

1. Методика оценки

Экзамен проводится в устной форме по билетам. Билет формируется по следующему правилу: один теоретический вопрос из списка и одна задача из списка. Задача реализуется в виде компьютерной программы, исполняемой на тестовых данных

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____
к экзамену по дисциплине «Программирование»

---

1. Статические элементы класса и методы класса и их использование. Множественное наследование. Виртуальные базовые классы.
2. Задача. Разработать указанный метод для заданного класса. Задача должна включать определение внутренних структур данных класса, разработку заголовка класса, конструктора и деструктора, указанного метода (функции или переопределенной операции с заданным способом передачи параметров или операндов), а также вспомогательных элементов класса, необходимых для реализации указанного метода (конструктор копирования, управление динамической памятью). Матрица переменной размерности, представленная динамическим массивом указателей на строки.

Утверждаю: зав. кафедрой ____ _____ должность, ФИО
(подпись)
(дата)

2. Критерии оценки

- Ответ на экзаменационный билет (тест) считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не написал текст программы для задачи,оценка составляет *<25% базовой.*

- Ответ на экзаменационный билет (тест) засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, но не раскрывает их содержимое, не может привести примеров использования конструкций языка, имеется текст программы, но она не работает или дает неправильный результат, оценка составляет 25-*50%базовой.*
- Ответ на экзаменационный билет (тест) билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, раскрывает их содержимое, анализирует примеры использования, но не объясняет отдельные особенности применения, программа компилируется и выполняется, но студент не может внести в нее изменения по требованию преподавателя, оценка составляет *50-75% базовой.*.
- Ответ на экзаменационный билет (тест) билет засчитывается на **продвинутом** уровне, если студент студент выполняет все перечисленные выше требования,
  оценка составляет *75-100% базовой.*.

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. **Вопросы к** экзамену **по дисциплине «**Программирование**»**

Теоретические вопросы

1. Тип данных и переменная. Базовые и производные типы данных. Иерархия определений типов данных и вложенности компонент переменных. Контекстный способ определения типа данных в Си. Абстрактный тип данных. Спецификация **typedef .**
2. Модульная организация программы. Время жизни и область действия переменных. Классификация. Определение и объявление переменных. Внешние, автоматические и статические переменные. Область действия функций. Внешние и статические функции.
3. Модульное программирование. Статическое связывание. Библиотеки. Заголовочные файлы, их назначение и содержание. Файл проекта в классическом программировании.
5. Указатели. Указатель как элемент архитектуры компьютера. Синтаксис указателя в Си. Указатель и ссылка. Передача формальных параметров и результата по значению и по ссылке. Адресная арифметика. Указатели и массивы. Способы работы через указатель с массивом. Динамическая память. Динамические переменные и массивы. Операторы и функции управления динамической памятью.
6. Трудоемкость алгоритмов. Определение трудоемкости. О-нотация. Способы оценки трудоемкости простейших программ.
7. Сортировка и поиск. Понятие записи и ключа. Линейный и двоичный поиск. Трудоемкость алгоритмов сортировки и поиска. Классификация сортировок: выбор, вставка, обмен, подсчет, разделение, слияние.
8. Структура данных как система взаимосвязанных переменных и значений. Статические и динамические структуры данных. Последовательность. Стек и

очередь. Свойства. Представление стека и очереди в массиве и списке. Использование стека при вызове функции.
9. Массивы указателей. Способы формирования массивов указателей - статические, динамические, смешанные. Работа с массивами указателей.
10. Списки. Определение элемента списка. Способы формирования списков. Односвязные списки. Двусвязные (циклические) списки.
11. Рекурсия. Рекурсивная структура данных и функция. Реализация рекурсивных функций, роль стека. Инвариант рекурсивной функции. Особенности разработки рекурсивных алгоритмов. Смысл локальных и глобальных переменных, формальных и фактических параметров в рекурсивной функции. Способы накопления результата.
12. Рекурсивные алгоритмы сортировки, трудоемкость. Рекурсивные алгоритмы комбинаторного перебора.
13. Рекурсивные поисковые алгоритмы. Сокращение пространства перебора. Жадные алгоритмы.
14. Деревья. Способы представления деревьев. Полный рекурсивный обход, ветвление. Алгоритмы, основанные на полном рекурсивном обходе дерева. Эффективность алгоритмов на деревьях.
15. Двоичное дерево. Основные характеристики и алгоритмы. Балансировка дерева.
16. Указатель на функцию. Его определение в языке и назначение. Указатель на функцию - формальный параметр. Динамическое связывание. Использование массивов указателей при реализации виртуальных функций.
17. Файл. Двоичный и текстовый файл. Запись. Последовательный и произвольный доступ. Текстовый файл. Особенности представления и преобразования строки текста. Функции для работы с текстовым файлом. Позиционирование в текстовом файле.
18. Двоичный файл. Способы распределения памяти в двоичном файле. Классификация форматов представления данных в двоичных файлах. Файлы записей фиксированной и переменной длины. Параметризованный файл записей фиксированной длины. Представление таблицы с произвольными типами полей в файле.
19. Указатель в файле. Связанные записи в файле. Сохранение и загрузка дерева и массива указателей в файле. Способы работы со связанными записями в файле: загрузка всей структуры данных, поэлементная загрузка, кэширование.
20. Определение класса и объекта. Свойства и методы (функции) класса. Синтаксическое и технологическое определение класса. Технология ООП. Программирование от функции к функции и от класса к классу. "Эпизодическое" и "тотальное" ООП.
21. Синтаксис определения класса и объекта в Си++. Функция, встраиваемые в структуру. Указательthis. Определение функции (метода) в заголовке класса и вне его.
22. Закрытая и общая части класса. Назначение закрытых и открытых данных и методов. Дружественность. Конструктор и деструктор. Глобальные, локальные и динамические объекты.
23. Переопределение операций. Синтаксис. Операнды, результат. Переопределение операций внутри и вне класса. Особенности переопределения некоторых операций (сравнение, присваивание, вывод в поток, [], (), приведение типа).
24. Конструктор копирования, принципы передачи объектов по ссылке и по значению. Способы передачи параметров (операндов) в методы. Конвейер ссылок и конвейер значений.

25. Основные технологические при принципы определения класса. Представление данных переменной размерности в объектах. Классы типов данных: строки переменной длины, матрицы, полиномы.
26. Представление структур данных в виде классов. Классы динамического массива указателей и списка и дерева.
27. Наследование. Иерархия объектов. Базовый и производный классы. Наследование данных и методов. Наследование как основа программирования "от класса к классу". Объектно-ориентированные библиотеки, работа с ними с использованием наследования. Наследование. Способы наследования методов: полное наследование, перекрытие, частичное (условное) наследование. Конструирование объектов вложенных классов.
28. Ограничения доступа при наследовании: личная (закрытая), общая и защищенная области класса. Обычное и публичное наследование.
29. Преобразование указателей на базовый и производный класс. "Расширение" и "сужение".Полиморфизм. Виртуальная функция. Определение и механизм реализации.
30. Внешний и внутренний полиморфизм. Виртуальная функция как основа для создания интерфейсов. Абстрактные базовые классы. Виртуальная функция как элемент "отложенного" программирования.
31. Статические элементы класса и методы класса и их использование. Множественное наследование. Виртуальные базовые классы.
32. Технология модульного проектирования ООП-программ. Статическое связывание и компоновка. Заголовочный файл и файл тела класса. Структура проекта ООП-программы.
33. Шаблон как макроопределение класса. Параметры шаблонов. Примеры шаблонов структур данных: стек, очередь, список. Требования к объектам - параметрам шаблонов.
34. Модульное программирование на "классическом Си". Статическое связывание: внешняя ссылка и точка входа. Заголовочные файлы, объектные модули, библиотека. Компоновка. Структура проекта на "классическом" Си
35. Модульное программирование на Си++. Файл описания класса и файл тела класса. Inline-методы. Структура проекта на Си++.

Экзаменационные задачи

Разработать указанный метод для заданного класса. Задача должна включать определение внутренних структур данных класса, разработку заголовка класса, конструктора и деструктора, указанного метода (функции или переопределенной операции с заданным способом передачи параметров или операндов), а также вспомогательных элементов класса, необходимых для реализации указанного метода (конструктор копирования, управление динамической памятью).

Содержание разрабатываемого класса.

1. Матрица переменной размерности, представленная динамическим массивом указателей на строки.
2. Матрица переменной размерности, представленная линейным динамическим массивом коэффициентов (двумерный массив разложен по строкам в линейный).
3. Разреженная матрица переменной размерности, представленная списком ненулевых коэффициентов (элемент списка содержит координаты коэффициента и его значение).
4. Разреженная матрица переменной размерности, представленная динамическим массивом описателей коэффициентов (описатель содержит координаты коэффициента и его значение).

5. Степенной полином произвольной степени.
6. Шаблон структуры данных – динамический массив указателей
7. Шаблон структуры данных – односвязный список, элемент списка содержит указатель на объект.
8. Шаблон структуры данных – односвязный список, элемент списка содержит непосредственно объект.
9. Шаблон структуры данных – двусвязный список, элемент списка содержит указатель на объект.
10. Шаблон структуры данных – двусвязный список, элемент списка содержит непосредственно объект.
11. Шаблон структуры данных – двусвязный циклический список, элемент списка содержит указатель на объект.
12. Шаблон структуры данных – двусвязный циклический список, элемент списка содержит непосредственно объект.
13. Шаблон структуры данных – двоичное дерево. Для операции извлечения по логическому номеру каждая вершина содержит счетчик вершин в поддереве.
14. Шаблон структуры данных – дерево, хранящее данные в концевых вершинах. Для операции извлечения по логическому номеру каждая вершина содержит счетчик вершин в поддереве.
15. Шаблон структуры данных – стек, представленный динамическим массивом объектов. При переполнении производится перераспределение памяти.
16. Шаблон структуры данных – стек, представленный односвязным списком.
17. Шаблон структуры данных – циклическая очередь, представленная динамическим массивом объектов. При переполнении производится перераспределение памяти.
18. Шаблон структуры данных – очередь, представленная циклическим списком.
19. Шаблон структуры данных – упорядоченная последовательность, представленная динамическим массивом объектов. При переполнении производится перераспределение памяти

Метод разрабатываемого класса.
1. Вставка по логическому номеру (переопределение операции **объект(T,int)**).
2. Вставка с сохранением порядка (переопределение операции **объект << T**).
3. Удаление по логическому номеру, переопределение операции **объект[int].**
4. Получение объекта по логическому номеру, переопределение операции **объект[int]**.
5. Добавление объекта (переопределение операции **объект << T**).
6. Переопределение операции вывода в текстовый поток (переопределение операции **ostream << объект**).
7. Переопределение операции ввода из текстового потока (переопределение операции **istream >> объект**).
8. Сохранение в двоичный файловый поток.
9. Загрузка из двоичного файлового потока.
10. Сортировка выбором.
11. Сортировка вставками.
12. Сложение двух объектов. Переопределение операции "+". Результат – новый объект (копия).
13. Умножение двух объектов. Переопределение операции "*". Результат – новый объект (копия).
14. Конструктор объекта из массива параметров (коэффициентов).
15. Конструктор копирования.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра автоматизированных систем управления
Кафедра автоматики
Кафедра вычислительной техники

Паспорт
курсовой работы

по дисциплине «Программирование», 3 семестр

1. **Методика оценки.**
Курсовой проект выполняется по одной из тем из предложенных тем в форме разработки пакета: пояснительная записка и программы (см. выбор темы).

Требования к содержанию и тестированию программы.

1. Программа должна быть проверена на входных данных различной размерности, например, на файлах различного размера. Необходимо фиксировать время выполнения программы, а также количество основных операций, например, сравнения. Строятся графики зависимостей этих параметров от характеристик файла (количество слов, размер). Производится оценка вида полученной зависимости (квадратичная, линейно-логирифмическая).
2. Программа должна отображать текущее состояние структуры данных, ее основные ее характеристики – количество элементов, процент их заполнения, разброс (минимальное и максимальное значение), время выполнения последней операции и т.п..

Требования к оформлению пояснительной записки:

- текст должен быть единообразно отформатирован и структурирован;
- основной текст – Times New Roman, шрифт 12;
- фрагменты программ – Arial или Courier. Текст программ должен быть отформатирован в соответствии с синтаксической структурой (вложенность фрагментов), иметь одинарный интервал между стоками (плотный текст);
- текст программы не должен содержать грамматических ошибок.

При невыполнении указанных требований записка возвращается на доработку.

Содержание пояснительной записки:

1. **Задание**.
2. **Структурное описание разработки** должно быть выполнено виде связного структурированного текста и давать разностороннее представление о программе: какие основные решения приняты при разработке, как работает программа, какие данные являются статическими, какие – динамическими и т.д.. Должны быть упомянуты особенности алгоритмов и дано их содержательное описание.
Необходимо использовать структурные схемы и рисунки. Особенно это касается того, что не может быть наблюдаемо непосредственно в тексте программы:
   - основные компоненты и связи между ними;
   - форматы входных и выходных данных, форматы внутренних структур данных;
   - содержательное описание алгоритмов работы на уровне компонент, связей и форматов данных.

3. **Функциональное описание**. Содержательное описание типов данных, переменных, интерфейсов функций, классов, фрагментов нетривиальных алгоритмов. Для описания рекомендуется использовать смешанное словесно/формальное представление программы с включением этих элементов в связный текст изложения материала.
4. **Описание работы программы** на контрольных примерах. Результаты измерений времени работы программы и количества основных операций для входных файлов различной размерности (не менее 4). Графики и оценка вида полученных зависимостей. Выводы. Ограничения (по памяти, по времени), ошибки, особенности проектирования.
5. **Приложение**: исходный текст программы с комментариями по существу алгоритма и структур данных.

В пояснительной записке структурное описание разработки должно давать представление о том, какие основные решения приняты при разработке, как работает программа, какие данные являются статическими, какие – динамическими, должны быть упомянуты особенности алгоритмов. Структурное описание – словесное, с привлечением графических иллюстраций (схем, рисунков), **фрагментов** оригинальных алгоритмов и структур данных.

2. **Критерии оценки и процедура защиты**

Для каждой группы вариантов устанавливается средний (начальный) уровень оценки, определяемый сложностью задания для данной дисциплины и семестра. При идеальном выполнении задания она может быть повышена, но не более, чем на балл.

Защита курсовой работы происходит в форме собеседования с вопросами как по пояснительной записке, так и по тексту программы. При желании можно получить предварительную консультацию по проблемам программирования или оформления пояснительной записки (при наличии соответствующих черновых материалов). Пояснительная записка сдается в виде оформленного документа (файл и твердая копия (без приложения)). Сдается также исходный текст программы.

3. Шкала оценки.

В общей оценке по дисциплине баллы за работы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. **Примерный перечень тем курсового проекта (работы).**

Форматирование гипертекстового (HTML) файла

1. Выходной файл представляет собой HTML-документ, с требуемым по заданию табличным представлением данных, либо наличием ссылок на выделяемые фрагменты. Входной файл представляет собой гипертекстовый документ (html-файл).При выполнении работы произвести измерение зависимости «грязного» времени работы программы и ее трудоемкости (количества базовых операций). Оценить вид полученной зависимости (линейно-логарифмическая, квадратичная).
2. Термином является слово, выделенное «жирным» шрифтом (тег). Программа находит термины в группе html-файлов, находящихся в заданном каталоге, и составляет таблицу ссылок на них, либо на абзацы, в которых они находятся.
3. Термином является слово, выделенное «жирным» шрифтом (тег).Программа находит термины в html-файле, при обнаружении того же

слова в последующем тексте формируется ссылка на термин, либо на его предыдущее упоминание (ссылку).

4. Программа открывает гипертекстовый документ и копирует его в заданный каталог. При обнаружении локальных гиперссылок, она также копирует документы, на которые они ссылаются.
5. Программа открывает гипертекстовый документ и копирует его в заданный каталог. При обнаружении локальных ссылок на изображения (теги типа <img>), она копирует соответствующие двоичные файлы.
6. Программа подсчитывает частоты появления слов в html-файле, слова считаются одинаковыми при совпадении первых 70% букв (% совпадения можно менять, синтаксис слов не учитывается). Формируется таблица из первых **n**наиболее часто встречающихся слов, в таблице содержится само слово и ссылки на все абзацы, где оно появляется.
7. Программа читает текст в html-файле, слова считаются одинаковыми при совпадении первых 70% букв (% совпадения можно менять, синтаксис слов не учитывается). При форматировании текста и обнаружении слова формируется ссылка на предыдущий абзац, где это слово встречается, либо на само слово.
8. Лексический контроль текста в html-файле. При чтении очередного слова в него «снимаются» окончания, а затем суффиксы. Полученная основа ищется в упорядоченном словаре и при ее отсутствии добавляется в словарь. Сам словарь хранится в виде отдельного текстового файла и для каждой основы содержит список найденных слов.

Иерархические структуры данных в памяти

1. На логическим уровне разрабатываемая структура данных представляет

собой обычную линейную последовательность элементов (например, строк) со стандартным набором операций (добавление в конец, вставка и удаление по логическому номеру, сортировка, бинарный поиск, вставка с сохранением порядка, сохранение и загрузка из текстового файла, выравнивание (балансировка - выравнивание размерностей структур данных нижнего уровня). Физическая структура данных имеет два уровня. На нижнем уровне поддерживается ограничение размерности структуры данных: при переполнении она разбивается пополам, соответствующие изменения вносятся в верхний уровень. При выполнении работы произвести измерение зависимости «грязного» времени работы программы и ее трудоемкости (количества базовых операций). Оценить вид полученной зависимости (линейно-логарифмическая, квадратичная).

2. Список - элемент содержит статический массив указателей на строки.

Включение с сохранением упорядоченности. Если после включения строки массив заполняется полностью, то создается еще один элемент списка с массивом указателей, в который переписывается половина указателей из старого.

3. Список, каждый элемент которого содержит динамический массив

указателей на строки, упорядоченные по длине. Размерность массива в каждом следующем элементе в 2 раза больше, чем в предыдущем. Строка включается в структуру данных с сохранением упорядоченности. Если строка включается в заполненный массив, то последний указатель перемещается в следующий элемент и так далее).

4. Двухуровневый массив указателей на строки. Массив верхнего уровня

статический, массивы нижнего уровня - динамические. Включение строки с сохранением упорядоченности. Если после включения строки массив заполняется полностью, то создается еще один массив указателей, в который переписывается половина указателей из старого.

5. Двухуровневый массив указателей на строки, упорядоченные по длине.

Размерность каждого следующего массива нижнего уровня в 2 раза больше предыдущего. Если строка включается в заполненный массив, то последний указатель перемещается в следующий элемент и так далее).

6. Массив указателей на заголовки списков. Элемент списка содержит

указатель на строку. (При включении последним предусмотреть ограничение длины текущего списка и переход к следующему). Текущие длины списков также необходимо хранить для исключения последовательного просмотра списков.

7. Список - каждый элемент является заголовком односвязного списка.

Элемент списка второго уровня содержит указатель на строку. (Включение элемента последним в список производить с учетом выравнивания длины текущего и следующего списков).

8. Односвязный (двусвязный) список. Для ускорения доступа к элементу

списка создается список верхнего уровня, для каждой группы из N элементов исходного списка создается элемент списка верхнего уровня с указателем на начало группы и числом элементов в ней. При вставке/удалении меняется «длина» группы, в которой производится операция.

9. Односвязный (двусвязный) список. Для ускорения доступа к элементу

списка создается массив указателей верхнего уровня, для каждой группы из N элементов исходного списка в нем имеется указатель на начало группы. Имеется также динамический массив «длин» групп. При вставке/удалении меняется «длина» группы, в которой производится операция.

Древовидные структуры

1. Для древовидных структур данных предусмотреть вывод характеристик

сбалансированности дерева (средняя длина ветви) и процедуру выравнивания (балансировки). При выполнении работы произвести измерение зависимости «грязного» времени работы программы и ее трудоемкости (количества базовых операций). Оценить вид полученной зависимости (линейно-логарифмическая, квадратичная).

2. Дерево, концевая вершина которого содержит внешний (динамический)

массив указателей постоянной размерности, а промежуточная - 2 указателя на потомков и счетчик вершин в поддереве. При переполнении концевой вершины она становится промежуточной и порождает два концевых потомка.

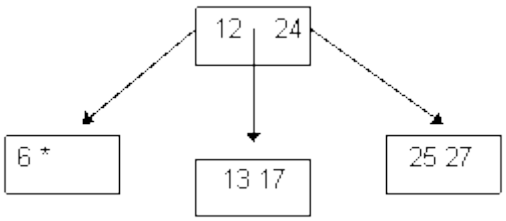
3. Двоичное дерево, каждая вершина которого содержит указатель на строку,

счетчик вершин в поддереве.

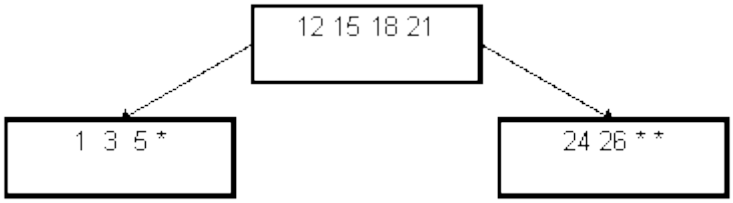
4. Двоичное дерево в динамическом массиве с вычисляемыми индексами

потомков (2n и 2n+1). Для хранении текста (слов) используется динамический массив указателей char**.

5. Упрощенное 2-3 дерево. Вершина дерева содержит два указателя на объекты и три указателя на поддеревья. Данные в дереве упорядочены.



6. Двоичное дерево с массивом объектов вершине. Вершина дерева содержит статический массив указателей на объекты и два указателя на правое и левое поддерево. Значения в дереве упорядочены. (массив в каждом элементе упорядочен, дерево в целом также упорядочено). Если при включении указателя в найденный массив последний переполняется, то самый правый указатель переносится в правое поддерево.



7. Дерево (иерархический каталог) объектов. Каждый каталог содержит список объектов (например, строк) и подкаталоги. Разработать операции создания пустого корневого каталога, позиционирования по дереву каталогов, добавление объектов и каталогов в текущий каталог, удаление объекта и каталога, а также поиск объекта по полному пути (parse). Структура данных – вершина с идентификатором типа (объект, каталог), именем и **динамическим массивом указателей на потомков**

8. Дерево (иерархический каталог) объектов. Каждый каталог содержит список объектов (например, строк) и подкаталоги. Разработать операции создания пустого корневого каталога, позиционирования по дереву каталогов, добавление объектов и каталогов в текущий каталог, удаление объекта и каталога, а также поиск объекта по полному пути (parse). Структура данных – вершина с идентификатором типа (объект, каталог), именем и **односвязным списком потомков.**

5. Перечень вопросов к защите курсового проекта (работы).

Защита курсовой работы происходит в форме собеседования с вопросами как по пояснительной записке, так и по тексту программы. При желании можно получить предварительную консультацию по проблемам программирования или оформления пояснительной записки (при наличии соответствующих черновых материалов). Пояснительная записка сдается в виде оформленного документа (файл и твердая копия (без приложения)). Сдается также исходный текст программы.