

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

Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ

к.т.н. Рева И. Л.

“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Программирование

Образовательная программа: 09.03.04 Программная инженерия, профиль: Технологии разработки программного обеспечения

Курс: 2, семестр : 3

Факультет автоматики и вычислительной техники,

		Семестр
№	Вид деятельности	3
1	Всего зачетных единиц (кредитов)	4
2	Всего часов	144
3	Всего занятий в контактной форме, час.	85
4	Лекции, час.	36
5	Практические занятия, час.	0
6	Лабораторные занятия, час.	36
7	из них в активной и интерактивной форме, час.	36
8	Аттестация, час.	2
9	Консультации, час.	11
10	Самостоятельная работа, час.	59
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	КР
12	Вид аттестации	Э

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.03.04 Программная инженерия

ФГОС введен в действие приказом №229 от 12.03.2015 г. , дата утверждения: 01.04.2015 г.

Место дисциплины в структуре учебного плана: Блок 1, базовая

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.03.04 Программная инженерия

Рабочая программа обсуждена на заседании кафедры

ВТ, протокол заседания кафедры №6 от 20.06.2017

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

Программу разработал:

доцент, к.т.н. Васюткина И. А.

Заведующий кафедрой:

доцент, к.т.н. Якименко А. А.

Ответственный за образовательную программу:

доцент Романов Е. Л.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ОПК.3 готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов; в части следующих результатов обучения:	
з3. знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения	
у1. владеть объектно-ориентированным подходом к программированию	
у6. владеть современными техническими и программными средствами взаимодействия с ЭВМ, технологиями разработки алгоритмов и программ, методами отладки и решения задач на ЭВМ в различных режимах	
Компетенция ФГОС: ОПК.4 способность осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий; в части следующих результатов обучения:	
з10. знать отечественные и международные стандарты в области документирования программного обеспечения	
Компетенция ФГОС: ПК.1 готовность применять основные методы и инструменты разработки программного обеспечения; в части следующих результатов обучения:	
з12. знать синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования	
з4. знать языки программирования и среды разработки	
з7. знать современные компиляторы, отладчики и оптимизаторы программного кода	
у4. уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	
у7. уметь применять методы и приемы отладки программного кода	
у8. уметь выявлять ошибки в программном коде	
у9. уметь применять выбранные языки программирования для написания программного кода	
Компетенция ФГОС: ПК.21 владение навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации; в части следующих результатов обучения:	
з1. знать методы повышения читаемости программного кода	

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.1.з12 знать синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования	
1. знать синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования	Лекции; Самостоятельная работа
ПК.1.з4 знать языки программирования и среды разработки	
2. знать языки программирования и среды разработки	Лабораторные работы; Самостоятельная работа
ПК.1.з7 знать современные компиляторы, отладчики и оптимизаторы программного кода	
3. знать современные компиляторы, отладчики и оптимизаторы программного кода	Лабораторные работы; Самостоятельная работа
ПК.1.у4 уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	
4. уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	Лабораторные работы; Самостоятельная работа

ПК.1.y7 уметь применять методы и приемы отладки программного кода	
5.уметь применять методы и приемы отладки программного кода	Лабораторные работы; Самостоятельная работа
ПК.1.y8 уметь выявлять ошибки в программном коде	
6.уметь выявлять ошибки в программном коде	Лабораторные работы; Самостоятельная работа
ПК.1.y9 уметь применять выбранные языки программирования для написания программного кода	
7.уметь применять выбранные языки программирования для написания программного кода	Лабораторные работы; Самостоятельная работа
ОПК.3.з3 знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения	
8.знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения	Лекции; Самостоятельная работа
ОПК.3.y1 владеть объектно-ориентированным подходом к программированию	
9.владеть объектно-ориентированным подходом к программированию	Лекции; Самостоятельная работа
ОПК.3.y6 владеть современными техническими и программными средствами взаимодействия с ЭВМ, технологиями разработки алгоритмов и программ, методами отладки и решения задач на ЭВМ в различных режимах	
10.владеть современными техническими и программными средствами взаимодействия с ЭВМ, технологиями разработки алгоритмов и программ, методами отладки и решения задач на ЭВМ в различных режимах	Самостоятельная работа
ОПК.4.з10 знать отечественные и международные стандарты в области документирования программного обеспечения	
11.знать отечественные и международные стандарты в области документирования программного обеспечения	Самостоятельная работа
ПК.21.з1 знать методы повышения читаемости программного кода	
12.знать методы повышения читаемости программного кода	Лабораторные работы; Самостоятельная работа

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 3				
Дидактическая единица: Расширения языка C				
1. Новые средства языка C++	0	3	1	слушает преподавателя, отвечает на вопросы
Дидактическая единица: Основные концепции объектно-ориентированного программирования				
2. Основные концепции объектно-ориентированного программирования: абстракция, инкапсуляция, наследование, полиморфизм	0	2	8	слушает, задает вопросы
Дидактическая единица: Основы объектно-ориентированного программирования на C++				
3. Классы. Члены класса. Модификаторы доступа к членам класса.	0	4	1, 9	слушает, задает вопросы

4. Переопределение операций языка С++ для применения к объектам классов. Дружественность.	0	6	1, 9	слушает, задает вопросы, отвечает на вопросы преподавателя
Дидактическая единица: Потоки ввода-вывода в С++				
5. Библиотека потоков ввода/вывода в С++. Перегрузка потоков ввода/вывода.	0	4	1, 9	слушает, задает вопросы
Дидактическая единица: Наследование в С++ . Полиморфизм				
6. Одиночное и множественное наследование в С++. Проблемы множественного наследования	0	4	1, 9	слушает, задает вопросы
7. Полиморфизм в классах. Виртуальные функции. Динамические структуры данных.	0	2	1, 9	слушает материал, задает вопросы
Дидактическая единица: Исключительные ситуации				
8. Исключения. Механизм обработки исключений. Библиотека классов исключений.	0	3	1, 9	слушает, задает вопросы
Дидактическая единица: Шаблоны функций и классов в С++				
9. Шаблоны функций и классов в С++ . Конкретизация шаблонов. Специализация шаблонов.	0	3	1, 9	слушает и задает вопросы
Дидактическая единица: Библиотеки языка С++. Класс String. RTTI. STL				
10. Библиотеки языка С++. Класс String. RTTI. Библиотека шаблонов STL	0	5	1, 9	слушает, задает вопросы

Таблица 3.2

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 3				
Дидактическая единица: Основы объектно-ориентированного программирования на С++				
1. Разработка классов, создание конструкторов и деструкторов. Использование статических членов класса	4	4	2, 3, 4, 5, 6, 7	изучает на практике разработку шаблона класса, выполняет индивидуальное задание
2. Переопределение операций	4	4	2, 3, 4, 5, 6, 7	изучает практически материал, выполняет индивидуальное задание
Дидактическая единица: Потоки ввода-вывода в С++				
3. Организация ввода-вывода в С++	4	4	2, 3, 4, 5, 6, 7	изучает на практике материал, выполняет индивидуальное задание
Дидактическая единица: Наследование в С++ . Полиморфизм				
4. Наследование	4	4	2, 3, 4, 5, 6, 7	изучает материал, выполняет индивидуальное задание
5. Полиморфизм. Создание динамического списка объектов, связанных наследованием	4	4	2, 3, 4, 5, 6, 7	изучает свойство ОО языков полиморфизм, выполняет индивидуальное задание
Дидактическая единица: Исключительные ситуации				

6. Обработка исключительных ситуаций	4	4	2, 3, 4, 5, 6, 7	изучает материал, выполняет индивидуальное задание
Дидактическая единица: Шаблоны функций и классов в C++				
7. Универсальность. Применение шаблонов функций и классов	4	4	2, 3, 4, 5, 6, 7	изучает материал, выполняет индивидуальное задание
Дидактическая единица: Библиотеки языка C++. Класс String. RTTI. STL				
8. Библиотека шаблонов STL.	4	4	12, 2, 3, 4, 5, 6, 7	изучает материал, выполняет индивидуальное задание
9. Заключительное занятие	4	4	2, 3, 4, 5, 6, 7	Защита лабораторной работы 8

Таблица 3.3

Темы для самостоятельного изучения	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 3				
Дидактическая единица: Изучение новых стандартов языка C++				
1. Стандарт языка C++ 14.	0	3	10, 11	Самостоятельно знакомится с новыми средствами языка C++, добавленными в стандарт 14

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 3				
1	Курсовая работа	10, 11, 12, 9	19	5
Выполнение индивидуального задания на курсовую работу: 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). - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000134024 . - Загл. с этикетки диска.- Рег. свидетельство №18891. Васюткина И. А. Программирование на языке высокого уровня [Электронный ресурс] : электронный учебно-методический комплекс / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156309 . - Загл. с экрана.				
2	Подготовка к занятиям	2, 3, 4, 5, 6, 7	9	4
Выполнение индивидуальных заданий по лабораторным работам, оформление отчетов: Васюткина И. А. Учебно-методические материалы по курсу «Программирование на языке высокого уровня» [Электронный ресурс] : [учебно-методическое пособие] / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000164070 . - Загл. с экрана. Васюткина И. А. Программирование на языке высокого уровня [Электронный ресурс] : электронный учебно-методический комплекс / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156309 . - Загл. с экрана.				
3	Дополнительная учебная деятельность	1, 8, 9	20	0
Изучение основной и дополнительной рекомендуемой литературы: 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). - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000134024 . - Загл. с этикетки диска.- Рег. свидетельство №18891. Васюткина И. А. Программирование на языке высокого уровня [Электронный ресурс] : электронный учебно-методический комплекс / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156309 . - Загл. с экрана.				
4	Подготовка к аттестации	1, 8, 9	8	2

подготовка к экзамену: 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). - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000134024. - Загл. с этикетки диска. - Рег. свидетельство №18891. Васюткина И. А. Программирование на языке высокого уровня [Электронный ресурс] : электронный учебно-методический комплекс / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156309. - Загл. с экрана.

5	Самостоятельное изучение теоретического материала	10, 11	3	0
---	---	--------	---	---

Студент изучает темы, приведенные в таблице 3.3 : 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). - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000134024. - Загл. с этикетки диска. - Рег. свидетельство №18891. Васюткина И. А. Программирование на языке высокого уровня [Электронный ресурс] : электронный учебно-методический комплекс / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156309. - 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	Портал НГТУ
Консультирование	e-mail
Контроль	Портал НГТУ
Размещение учебных материалов	e-mail:сервер кафедры ВТ

Таблица 5.2

Активные и интерактивные формы проведения занятий

№	Наименование активных форм	Коды формируемых компетенций
1	Метод проектов	ПК.1;
Формируемые умения: 34. знать языки программирования и среды разработки		
Краткое описание применения: Самостоятельно выполняется практическая задача, изучается материал дидактической единицы.		

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставить оценки по традиционной шкале и 15-уровневой ECTS.

Краткая информация о БРС приведена в табл. 6.1.

Подробная информация о БРС приводится в Приложении №1 к рабочей программе.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл
Семестр: 3		

Лабораторная №1: Выполнение	2	5
Лабораторная №1: Защита	2	3
Лабораторная №2: Выполнение	2	4
Лабораторная №2: Защита	1	3
Лабораторная №3: Выполнение	2	4
Лабораторная №3: Защита	1	3
Лабораторная №4: Выполнение	2	5
Лабораторная №4: Защита	2	3
Лабораторная №5: Выполнение	2	5
Лабораторная №5: Защита	2	3
Лабораторная №6: Выполнение	2	4
Лабораторная №6: Защита	1	3
Лабораторная №7: Выполнение	2	5
Лабораторная №7: Защита	2	3
Лабораторная №8: Выполнение	2	4
Лабораторная №8: Защита	1	3
Курсовая работа: Итого	0	100
Контролирующие материалы приводятся в "Васюткина И. А. Программирование на языке высокого уровня [Электронный ресурс] : электронный учебно-методический комплекс / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156309 . - Загл. с экрана."		
Курсовая работа: Проектирование и реализация	0	40 (в состав баллов за КР)
Курсовая работа: Оформление отчета	0	20 (в состав баллов за КР)
Курсовая работа: Защита	0	40 (в состав баллов за КР)
Экзамен:	20	40
Контролирующие материалы приводятся в "Васюткина И. А. Программирование на языке высокого уровня [Электронный ресурс] : электронный учебно-методический комплекс / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156309 . - 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 КП/КР	Экзамен
ОПК.3	з3. знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения		+
	у1. владеть объектно-ориентированным подходом к программированию		+
	уб. владеть современными техническими и программными средствами взаимодействия с ЭВМ, технологиями разработки алгоритмов и программ, методами отладки и решения задач на ЭВМ в различных режимах	+	
ОПК.4	з10. знать отечественные и международные стандарты в области документирования программного обеспечения	+	
ПК.1	з12. знать синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования		+

	34. знать языки программирования и среды разработки	+	
	37. знать современные компиляторы, отладчики и оптимизаторы программного кода	+	
	у4. уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	+	
	у7. уметь применять методы и приемы отладки программного кода	+	
	у8. уметь выявлять ошибки в программном коде	+	
	у9. уметь применять выбранные языки программирования для написания программного кода		+
ПК.21	31. знать методы повышения читаемости программного кода	+	

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

7. Литература

Основная литература

1. 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.
2. Хабибуллин И. . Программирование на языке высокого уровня C/C++ : учебное пособие для вузов по направлению 654600 "Информатика и вычислительная техника" / И. Ш. Хабибуллин. - СПб, 2006. - 485 с. : ил.
3. Дейтел Х. М. Как программировать на С++ / Х. М. Дейтел, П. Дж. Дейтел ; пер. с англ. под ред. В. В. Тимофеева. - М., 2007. - 799 с. : ил.
4. Сеницын С. В. Программирование на языке высокого уровня : учебник / С. В. Сеницын, А. С. Михайлов, О. И. Хлытчиев. - М., 2010. - 392, [1] с. : ил., табл.
5. Павловская Т. А. C/C++. Программирование на языке высокого уровня : [учебник по направлению "Информатика и вычислительная техника"] / Т. А. Павловская. - СПб. [и др.], 2010. - 460 с. : табл., ил.
6. 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). - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000134024. - Загл. с этикетки диска. - Рег. свидетельство №18891.

Дополнительная литература

1. Шилдт Г. С++ : для начинающих : самоучитель / Герберт Шилдт ; пер. с англ. К. Г. Финогенова. - М., 2007. - 639 с. : ил.
2. Павловская Т. А. C/C++ Программирование на языке высокого уровня : издательская программа 300 лучших учебников для высшей школы в честь СПб. / Т. А. Павловская. - М. [и др.], 2005. - 460 с. : ил.
3. Шилдт Г. Искусство программирования на С++ : [для программистов] / Герберт Шилдт ; [пер. с англ. Т. Коротяевой]. - СПб., 2005. - 474 с. : ил.

Интернет-ресурсы

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znaniy.com" : <http://znaniy.com/>

5. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Васюткина И. А. Учебно-методические материалы по курсу «Программирование на языке высокого уровня» [Электронный ресурс] : [учебно-методическое пособие] / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000164070. - Загл. с экрана.
2. Васюткина И. А. Программирование на языке высокого уровня [Электронный ресурс] : электронный учебно-методический комплекс / И. А. Васюткина ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156309. - Загл. с экрана.

8.2 Специализированное программное обеспечение

- 1 Microsoft Windows
- 2 Visual Studio

9. Материально-техническое обеспечение

Презентационное оборудование

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Для изложения лекционного материала разработаны презентации

Компьютерный класс

№	Наименование	Назначение
1	Компьютерный класс (Компьютеры объединены в локальную сеть с выходом в Internet)	Задания лаб. работ выполняются на компьютерах

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

Кафедра вычислительной техники

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

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Программирование

Образовательная программа: 09.03.04 Программная инженерия, профиль: Технологии разработки программного обеспечения

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.3 готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	з3. знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения	Основные концепции объектно-ориентированного программирования: абстракция, инкапсуляция, наследование, полиморфизм		Экзамен, вопрос 1
ОПК.3	у1. владеть объектно-ориентированным подходом к программированию	Библиотека потоков ввода/вывода в C++. Перегрузка потоков ввода/вывода. Библиотеки языка C++. Класс String, RTTI. Библиотека шаблонов STL Исключения. Механизм обработки исключений. Библиотека классов исключений. Классы. Члены класса. Модификаторы доступа к членам класса. Одиночное и множественное наследование в C++. Проблемы множественного наследования Переопределение операций языка C++ для применения к объектам классов. Дружественность. Полиморфизм в классах. Виртуальные функции. Динамические структуры данных. Шаблоны функций и классов в C++ . Конкретизация шаблонов. Специализация шаблонов.		Экзамен, вопросы 2-30
ОПК.3	уб. владеть современными техническими и программными средствами взаимодействия с ЭВМ, технологиями разработки алгоритмов и программ, методами отладки и решения задач на ЭВМ в различных режимах	Стандарт языка C++ 14.	Курсовая работа, этап 2-3	

ОПК.4 способность осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	з10. знать отечественные и международные стандарты в области документирования программного обеспечения	Стандарт языка C++ 14.	Курсовая работа, этап 4.	
ПК.1/ПТ готовность применять основные методы и инструменты разработки программного обеспечения	з12. знать синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования	Библиотека потоков ввода/вывода в C++. Перегрузка потоков ввода/вывода. Библиотеки языка C++. Класс String. RTTI. Библиотека шаблонов STL Исключения. Механизм обработки исключений. Библиотека классов исключений. Классы. Члены класса. Модификаторы доступа к членам класса. Новые средства языка C++ Одночное и множественное наследование в C++. Проблемы множественного наследования Переопределение операций языка C++ для применения к объектам классов. Дружественность. Полиморфизм в классах. Виртуальные функции. Динамические структуры данных. Шаблоны функций и классов в C++ . Конкретизация шаблонов. Специализация шаблонов.		Экзамен, вопросы 2-30
ПК.1/ПТ	з4. знать языки программирования и среды разработки	Библиотека шаблонов STL. Заключительное занятие Наследование Обработка исключительных ситуаций Организация ввода-вывода в C++ Переопределение операций Полиморфизм. Создание динамического списка объектов, связанных наследованием Разработка классов, создание конструкторов и деструкторов. Использование статических членов класса Универсальность. Применение шаблонов функций и классов	Курсовая работа, этапы 1-2	
ПК.1/ПТ	з7. знать современные компиляторы, отладчики и оптимизаторы программного кода	Библиотека шаблонов STL. Заключительное занятие Наследование Обработка исключительных ситуаций Организация ввода-вывода в C++ Переопределение операций Полиморфизм. Создание динамического списка объектов, связанных	Курсовая работа, этапы 2-3	

		наследованием Разработка классов, создание конструкторов и деструкторов. Использование статических членов класса Универсальность. Применение шаблонов функций и классов		
ПК.1/ПТ	у4. уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	Библиотека шаблонов STL. Наследование Обработка исключительных ситуаций Организация ввода-вывода в C++ Переопределение операций Полиморфизм. Создание динамического списка объектов, связанных наследованием Разработка классов, создание конструкторов и деструкторов. Использование статических членов класса Универсальность. Применение шаблонов функций и классов	Курсовая работа, этапы 2-3	
ПК.1/ПТ	у7. уметь применять методы и приемы отладки программного кода	Библиотека шаблонов STL. Наследование Обработка исключительных ситуаций Организация ввода-вывода в C++ Переопределение операций Полиморфизм. Создание динамического списка объектов, связанных наследованием Разработка классов, создание конструкторов и деструкторов. Использование статических членов класса Универсальность. Применение шаблонов функций и классов	Курсовая работа, этап 3	
ПК.1/ПТ	у8. уметь выявлять ошибки в программном коде	Библиотека шаблонов STL. Наследование Обработка исключительных ситуаций Организация ввода-вывода в C++ Переопределение операций Полиморфизм. Создание динамического списка объектов, связанных наследованием Разработка классов, создание конструкторов и деструкторов. Использование статических членов класса Универсальность. Применение шаблонов функций и классов	Курсовая работа, этап 3	
ПК.1/ПТ	у9. уметь применять выбранные языки программирования для написания программного кода	Библиотека шаблонов STL. Наследование Обработка исключительных ситуаций Организация ввода-вывода в C++ Переопределение операций Полиморфизм. Создание динамического списка объектов, связанных наследованием Разработка классов, создание конструкторов и деструкторов. Использование статических членов класса Универсальность. Применение шаблонов функций и классов		Экзамен, вопросы 2-30

ПК.21/П владение навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации	31. знать методы повышения читаемости программного кода	Библиотека шаблонов STL.	Курсовая работа, этапы 2-3	
--	---	--------------------------	----------------------------	--

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.3, ОПК.4, ПК.1/ПТ, ПК.21/П.

Экзамен проводится в устной форме по билетам. Билет содержит один теоретический вопрос и одну задачу: вопрос выбирается из диапазона вопросов (список вопросов приведен в паспорте экзамена), задача также выбирается из списка задач и решается студентом в письменном виде.

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 дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.3, ОПК.4, ПК.1/ПТ, ПК.21/П, 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.

Паспорт экзамена

по дисциплине «Программирование», 3 семестр

1. Методика оценки

Экзамен проводится в устной форме, по билетам. Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов (список вопросов приведен ниже). Вторым пунктом в билете является задача, выбираемая из списка задач и решаемая студентом в письменном виде. В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к экзамену по дисциплине «Программирование»

1. Проблемы множественного наследования. Виртуальные классы при множественном наследовании.
2. Задача.

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

1. Критерии оценки

- Ответ на экзаменационный билет считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен проиллюстрировать связь теории с практическим применением, при решении задачи допускает принципиальные ошибки, оценка составляет 0-20 баллов.
- Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может применять средства языка программирования при решении практических задач, но ответ недостаточно полон, при решении задачи допускает непринципиальные ошибки, например, синтаксические, оценка составляет 21 - 29 баллов.
- Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, может показать и объяснить

применение методов и средств языка при решении практических задач, легко ориентируется в изученном материале, ответ достаточно полон, не допускает ошибок при решении задачи, оценка составляет 30 - 35 баллов.

- Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы четко и грамотно формулирует основные понятия, способен анализировать и принимать эффективные решения поставленных задач, приводит конкретные примеры из практики, не допускает ошибок и способен обосновать выбор метода решения задачи, оценка составляет 36 - 40 баллов.

2. Шкала оценки

Общая оценка за экзамен по дисциплине складывается из баллов, полученных при выполнении лабораторных работ в течение семестра и на экзамене.

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

Итоговая оценка по зачету выставляется по 100-балльной шкале, по буквенной шкале ECTS и в традиционной форме (в соответствии с действующим Положением о балльно-рейтинговой системе оценки достижений студентов НГТУ).

3. Вопросы к экзамену по дисциплине «Программирование»

1. Основные свойства объектно-ориентированного программирования.
2. Новые средства языка C++.
3. Понятие класса в C++ и его роль в ООП. Описание класса в C++. Члены класса.
4. Создание и использование объектов в C++. Указатели на объекты класса. Массивы объектов. Вызов методов класса. Указатель this.
5. Конструкторы. Конструктор копирования. Деструктор.
6. Статические члены класса. Описание, определение и инициализация.
7. Модификаторы доступа к членам класса. Влияние модификаторов на доступ к членам класса.
8. Дружественные функции и классы.
9. Перегрузка унарных операций. Способы перегрузки. Примеры перегрузки унарных операций: вызова функции и индексирования.
10. Перегрузка унарных операций. Способы перегрузки. Примеры перегрузки операций: присваивания и инкремента и декремента.
11. Перегрузка бинарных операций. Способы перегрузки. Примеры перегрузки бинарных арифметических операций.
12. Перегрузка операций ввода/вывода в стандартный и файловый поток.
13. Наследование. Модификаторы доступа при наследовании и их влияние на доступ к членам родительского класса.
14. Одиночное наследование. Конструкторы и деструкторы. Порядок создания и удаления объекта наследника.
15. Множественное наследование. Конструкторы и деструкторы. Порядок создания и удаления объекта наследника.
16. Проблемы множественного наследования. Виртуальные классы при множественном наследовании.
17. Виртуальные методы и деструкторы. Понятие полиморфизма. Реализация механизма полиморфизма.
18. Отношение включения и отношение наследования. Локальные и вложенные классы.
19. Понятие потока ввода/вывода. Виды потоков. Иерархия классов-потоков.

20. Форматирование в потоках ввода/вывода.
21. Состояние потока ввода/вывода. Обработка ошибок в потоках.
22. Создание, чтение и запись текстовых файлов +-
23. Создание, чтение и запись двоичных файлов.
24. Исключения в C++. Механизм обработки исключительных ситуаций.
25. Обработка исключительных ситуаций в классах. Классы исключений.

Стандартная обработка исключений.

26. Шаблоны функций в C++. Специализированные шаблоны функций.
27. Шаблоны классов в C++. Определение функций шаблонных классов.

Специализированные шаблоны классов.

28. Последовательные контейнеры. Организация хранения данных и методы их обработки.

29. Ассоциативные контейнеры. Организация хранения данных и методы их обработки.

30. Приведение типов в C++. Механизм RTTI.

Примеры задач экзамена

Задача 1.

Класс разреженных матриц. Матрица представлена динамическим массивом ненулевых коэффициентов (строка, столбец - int, значение - double). Переопределить операцию сложения матриц, результат - объект-значение, операнды не меняются.

Задача 2.

Шаблон класса – двусвязный список, хранит элементы типа T. Переопределить операции вставки элемента по логическому номеру, сортировки списка.

Задача 3.

Шаблон класса – двухсвязный список. Каждый элемент списка содержит статический массив указателей на элементы типа T. Первый пустой указатель в массиве имеет значения NULL. Реализовать операцию удаления элемента по логическому номеру. Освободившийся массив указателей удаляется из списка.

Паспорт курсовой работы

по дисциплине «Программирование», 3 семестр

1. Методика оценки.

Задание на курсовую работу выдается каждому студенту индивидуально. Курсовая работа должна быть выполнена с использованием технологии объектно-ориентированного программирования (классы, наследование, виртуальные функции, шаблоны) и реализована в виде набора классов. Программа демонстрирует выполнение всех своих функций в любой последовательности через систему меню.

Структура:

1. Проектирование и реализация системы классов.
2. Проверка на тестовых данных.
3. Пояснительная записка.

Этапы выполнения и защиты:

1. Проектирование структур данных.
2. Реализация функциональности классов.
3. Тестирование.
4. Написание пояснительной записки к курсовой работе.
5. Защита курсовой работы.

Оцениваемые позиции:

1. Реализация в программе заданной функциональности.
2. Ответы на поставленные вопросы по работе.
3. Качество оформления пояснительной записки.

Правила оформления пояснительной записки к курсовой работе:

Пояснительная записка должна иметь объем не менее 20 стр. текста (без приложения, шрифт 12), выполнена в соответствии с ГОСТом и содержать следующие разделы.

1. Развернутое описание задания.
2. Структурное описание разработки. Описание используемых структур данных, форматов данных, сопровождаемое необходимыми графическими иллюстрациями (рисунками). Содержательное описание основных алгоритмов, их особенностей (4-5 стр.).
3. Функциональное описание. Содержательное описание типов данных, структур данных (шаблоны классов), фрагментов нетривиальных алгоритмов. Для описания использовать смешанное текстово-формальное представление программы с включением этих элементов в связный текст изложения материала.
4. Описание пользовательского интерфейса.

5. Описание работы программы на контрольных примерах (например, на большом текстовом файле), временные характеристики программы на простых типах для одного из методов: сортировка, балансировка, поиск в неупорядоченной последовательности.
6. Выводы. Описание того, что было сделано. Ограничения (по памяти, по времени), ошибки, особенности проектирования.
7. Список используемой литературы.
8. Приложение: текст программы с комментариями по существу алгоритма и структур данных.

2. Критерии оценки.

- работа считается **не выполненной**, если реализован не весь функционал программы, указанный в задании, при защите студент не может объяснить алгоритмы работы программы, пояснительная записка выполнена небрежно и содержит ошибки, оценка составляет 0-49 баллов.
- работа считается выполненной **на пороговом** уровне, если реализован весь функционал программы, указанный в задании, при защите студент объясняет алгоритмы работы программы с неточностями, есть небольшие недочеты в программе, а также пояснительная записка содержит некоторое количество ошибок, оценка составляет 50 - 72 баллов.
- работа считается выполненной **на базовом** уровне, если реализован весь функционал программы, указанный в задании, при защите студент грамотно объясняет алгоритмы работы программы, отвечает на поставленные вопросы, пояснительная записка выполнена согласно требованиям по оформлению, оценка составляет 73-86 баллов.
- работа считается выполненной **на продвинутом** уровне, если реализован весь функционал программы, указанный в задании, студент в разработке применил оригинальные подходы и алгоритмы, при защите студент отвечает на все поставленные вопросы полно, пояснительная записка выполнена согласно требованиям по оформлению, оценка составляет 87-100 баллов.

3. Шкала оценки.

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

Курсовая работа оценивается по 100 балльной системе в соответствии с правилами балльно-рейтинговой системы.

4. Примерный перечень тем курсового проекта (работы).

1. Шаблон иерархической структуры данных в памяти

Для заданной двухуровневой структуры данных, содержащей указатели на объекты (или сами объекты) - параметры шаблона, разработать полный набор операций (добавление, включение и извлечение по логическому номеру, сортировка, включение с сохранением порядка, загрузка и сохранение строк в бинарном файле, балансировка – выравнивание размерностей структур данных нижнего уровня). Предполагается, что операции сравнения хранимых объектов переопределены стандартным образом (в виде операций <, > и т.д.). Программа должна использовать шаблонный класс с объектами-строками и реализовывать указанные выше действия над текстом любого объема, загружаемого из файла.

Программа должна реализовывать указанные выше действия. Протестировать структуру данных. Программа тестирования должна содержать **меню**, обеспечивающее выбор операций.

1. Шаблон структуры данных – двухуровневый массив указателей на объекты. Массив верхнего уровня – статический, массивы нижнего уровня – динамические, размерность – параметр конструктора, последовательность указателей в каждом массиве ограничена NULL. Если после включения указателя массив заполняется полностью, то создается еще один массив указателей, в который переписывается половина указателей из старого.

2. Шаблон структуры данных – двухуровневый массив указателей на объекты с типом – параметром шаблона. Массив верхнего уровня – статический. Массивы нижнего уровня – динамические, размерность каждого следующего в 2 раза больше размерности предыдущего, последовательность указателей в каждом массиве ограничена NULL.

3. Шаблон структуры данных – односвязный список, каждый элемент которого содержит статический массив указателей на объекты. Последовательность указателей в каждом массиве ограничена NULL. При переполнении текущего массива указателей создается новый элемент списка, в который переписывается половина указателей из текущего.

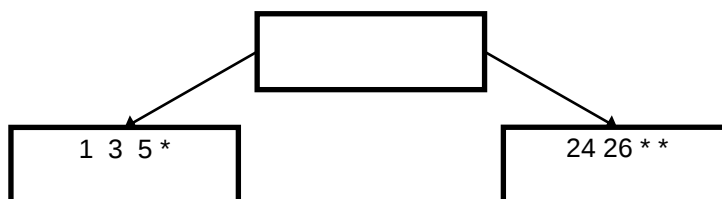
4. Шаблон структуры данных – односвязный список, каждый элемент которого содержит динамический массив указателей на объекты. Размерность массива указателей в каждом последующем элементе списка в 2 раза больше, чем в предыдущем. Последовательность указателей в каждом массиве ограничена NULL. При переполнении текущего массива указателей последний указатель переносится в следующий элемент списка.

5. Шаблон структуры данных – двусвязный циклический список, каждый элемент которого содержит статический массив указателей на объекты. Последовательность указателей в каждом массиве ограничена NULL. При переполнении текущего массива указателей создается новый элемент списка, в который переписывается половина указателей из текущего.

6. Шаблон структуры данных – односвязный список, каждый элемент является заголовком односвязного списка. Элемент списка второго уровня содержит указатель на объект. (При включении элемента последним в список предусмотреть ограничение длины текущего списка и переход к следующему).

7. Шаблон структуры данных – массив указателей на заголовки списков. Элемент списка содержит указатель на объект. (При включении элемента последним в список предусмотреть ограничение длины текущего списка и переход к следующему).

8. Шаблон структуры данных – дерево. Конечная вершина дерева содержит статический массив указателей на объекты (ограниченный NULL-указателем), промежуточная вершина содержит счетчик вершин дерева и два указателя на правое и левое поддерево. Значения в дереве не упорядочены. Естественная нумерация значений производится путем обхода дерева по принципу “левое поддерево – вершина – правое поддерево”. Если при включении указателя в найденный массив последний переполняется, он становится промежуточной вершиной.



9. Шаблон структуры данных – двусвязный список (нециклический), каждый элемент списка содержит указатель на объект. Для ускорения процедуры обхода структуры данные имеются динамический массив указателей на каждый 10-ый элемент списка (0,10,20 и т.д.).

2. Класс - структура данных в двоичном файле

Класс двоичного файла, производный от `fstream`. Двоичный файл содержит заданную структуру данных. Программа должна представлять собой простой текстовый редактор, использующий структуру данных для промежуточного хранения редактируемого файла. Должны быть реализованы операции создания и заполнения двоичного файла из заданного текстового и сохранения содержимого двоичного файла в текстовый.

Программа должна реализовывать указанные выше действия. Протестировать структуру данных. Программа тестирования должна содержать **меню**, обеспечивающее выбор операций.

1. Класс – двоичный файл, производный от `fstream`. Файл содержит массив указателей на строки, представленные записями переменной длины: целый счетчик и последовательность символов строки (без 0-ограничителя). Формат файла: в начале – файла, размерность массива указателей, текущее количество указателей, адрес (смещение) массива указателей в файле.
2. Класс – двоичный файл, производный от `fstream`. Файл содержит односвязный список строк в формате записей переменной длины со счетчиком. Формат файла: в начале файла – заголовок списка. Элемент списка содержит файловый указатель на следующий и саму строку в виде записи переменной длины.
3. Класс – двоичный файл, производный от `fstream`. Файл содержит двусвязный циклический список строк в формате записей переменной длины со счетчиком. Формат файла: в начале файла – заголовок списка. Элемент списка содержит файловые указатели на следующий и предыдущий и саму строку в виде записи переменной длины.
4. Класс – двоичный файл, производный от `fstream`. Файл содержит дерево, конечная вершина которого содержит строку, представленную записью переменной длины: целый счетчик и последовательность символов строки (без 0-ограничителя). Промежуточная вершина содержит указатели на правое и левое поддерево и число вершин в поддереве. Формат файла: в начале – файла, указатель на корневую вершину.

3. Шаблон - структура данных в двоичном файле

Класс двоичного файла, производный от `fstream`. Двоичный файл содержит заданную структуру данных с типом хранимых объектов - параметром шаблона. Программа должна выполнять операции создания файла, просмотра, добавления, удаления, обновления и сортировки объектов заданного при генерации типа. Реализовать две версии программы для различных хранимых объектов (например, текстовых строк и объектов класса из лабораторной работы). Предполагается, что операции сравнения хранимых объектов переопределены стандартным образом (в виде операций `<`, `>` и т.д.). Для позиционирования в потоке можно использовать методы `seekg(long,mode)` и `long tellg()`, для хранимого объекта переопределен метод `fstream << object` который записывает объект с текущей позиции в открытый двоичный поток и метод `fstream >> object`, который читает содержимое объекта с текущей позиции открытого потока.

Программа должна реализовывать указанные выше действия. Протестировать структуру данных на простом типе данных (например, int, double и т.д.) и сложном. Программа тестирования должна содержать **меню**, обеспечивающее выбор операций.

1. Шаблон – двоичный файл, содержащий двусвязный циклический список объектов. Тип хранимого в файле объекта – параметр шаблона. В начале файла расположен файловый указатель на первый элемент списка. Элемент списка содержит 2 файловых указателя на следующий и предыдущий элементы, за которыми следует объект.
2. Шаблон – двоичный файл, содержащий односвязный список объектов. Тип хранимого в файле объекта – параметр шаблона. В начале файла расположен файловый указатель на первый элемент списка. Элемент списка содержит файловый указатель на следующий элемент, за которым следует объект.
3. Шаблон – двоичный файл, содержащий массив указателей на объекты. Тип хранимого в файле объекта – параметр шаблона. В начале файла расположены: размерность массива указателей (int), текущее количество указателей (int) и смещение (адрес) массива указателей. Если происходит заполнение массива указателей, то его размерность увеличивается в 2 раза и он переписывается в конец файла.
4. Шаблон – двоичный файл, содержащий двоичное дерево объектов. Тип хранимого в файле объекта – параметр шаблона. В начале файла расположен указатель на корневую вершину. Вершина содержит два файловых указателя на правое и левое поддерево и сам объект, который непосредственно следует за указателями.
5. Шаблон – двоичный файл, содержащий двоичное дерево объектов. Тип хранимого в файле объекта – параметр шаблона. В начале файла расположен указатель на корневую вершину. Вершина содержит два файловых указателя на правое и левое поддерево, счетчик количества объектов (int) и статический массив указателей на объекты. Структура данных упорядочена (дерево и массив объектов).

4. Коллекция объектов произвольных типов в памяти

Необходимо разработать интерфейс для объединения в структуру данных множества объектов различных классов - абстрактный базовый класс объектов object, для которого предусмотреть виртуальные методы: загрузки объекта из текстовой строки, выгрузки объекта в текстовую строку в динамической памяти, добавления объекта в последовательный двоичный файл, чтения объекта из последовательного двоичного файла, возврата уникального идентификатора класса, возврата указателя на строку с именем класса, сравнения двух объектов, “сложения” (“объединения”) двух объектов, создание динамической копии объекта. Сделать классы хранимых объектов производными от абстрактного базового класса object (например, класс Float, String и класс из лабораторной работы).

Коллекция представляет собой структуру данных, хранящую указатели на объекты произвольного типа - объекты базового класса. Программа должна обеспечивать просмотр коллекции, добавление объекта выбранного типа, редактирование и удаление объекта, сложение и сравнение объектов при условии совпадения типов с помещением результата в ту же коллекцию, сохранение и загрузку объектов из текстового (или двоичного) файла.

Программа должна реализовывать указанные выше действия. Протестировать структуру данных на простом типе (например, Int или String). Программа тестирования должна содержать **меню**, обеспечивающее выбор операций.

Вид структуры данных

1. Двусвязный список.
2. Двусвязный циклический список.

3. Список с организацией стека.
4. Список с организацией очереди.
5. Дерево, вершина которого содержит два указателя на поддеревья, счетчик количества вершин и указатель на объект.

Дополнение: добавить в коллекцию вложенный класс, реализующий функции итератора (дополнительный балл 0,5).

5. Реализовать АТД «Файловая структура данных» в соответствии с вариантом.

Операции АТД:

Insert – вставка записи в файл,

Delete – удаление записи из файла,

Search - поиск записи в файле.

1. Элементом данных, хранящихся в файле, является запись, имеющая уникальное ключевое значение.
2. Запись в файле представлена индексом, т.е. парой (k, p) , где k – ключевое значение, p – файловый указатель на начало записи в файле.
3. Для файла данных поддерживается файловая структура индексов, указанная в варианте задания.
4. Файл индексов имеет страничную структуру. Страницы содержат индексы записей и имеют фиксированный размер.
5. Чтение и запись в файл индексов ведется постранично.
6. Тестирование файловой структуры ведется для различных значений параметров:
 N – число записей в файле данных, $N = 10^3, 10^4, 10^5, 10^6$,
 M – число индексов на странице файла индексов, $M = 10, 100, 1000$.
7. Число обращений к блокам файла индексов в процессе выполнения операций должно соответствовать:
 - для хешированного файла - $N/(M \cdot K)$, где K – число сегментов хеш – таблицы,
 - для индексированного файла – $2 + \log(N/M)$,
 - для В- дерева файла – $2 + \log_t(N/M)$, где t – степень В – дерева.

Варианты заданий.

1. АТД «Плотный индекс файла». Записи не закрепленные. Ключ записи – строка символов переменной длины.
2. АТД «Плотный индекс файла». Записи закрепленные. Ключ записи – строка символов переменной длины.
3. АТД «Плотный индекс файла». Записи не закрепленные. Ключ записи – вещественное число.
4. АТД «Плотный индекс файла». Записи закрепленные. Ключ записи - вещественное число.
5. АТД «Разреженный индекс файла». Записи не закрепленные. Ключ записи – строка символов переменной длины.
6. АТД «Разреженный индекс файла». Записи не закрепленные. Ключ записи – вещественное число.
7. АТД «Разреженный индекс файла». Записи не закрепленные. Ключ записи – вещественное число. Блок файла индексов имеет $K = M/10$ для возможности дальнейшей вставки записей без выделения новых страниц индексов.
8. АТД «В – дерево файла» с нисходящим разделением полных узлов при вставке записей. Записи не закрепленные. Ключ записи – целое число.
9. АТД «В – дерево файла» с нисходящим разделением полных узлов при вставке записей. Записи закрепленные. Ключ записи – целое число.

10. АД «В – дерево файла» с восходящим разделением полных узлов при вставке записей. Записи не закрепленные. Ключ записи – целое число.
11. АД «В – дерево файла» с восходящим разделением полных узлов при вставке записей. Записи закрепленные. Ключ записи – целое число.
12. АД «Хешированный файл». Метод хеширования – метод цепочек. Записи не закрепленные. Ключ записи – строка символов переменной длины.
13. АД «Хешированный файл». Метод хеширования – метод цепочек. Записи закрепленные. Ключ записи – строка символов переменной длины.

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

При защите курсовой работы вопросы студент отвечает на вопросы по реализуемым структурам данных и алгоритмам.