Частное учреждение образования МИНСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ

Кафедра автоматизированных информационных систем

УТВ	ВЕРЖДАЮ	
Дек	ан учётно-	финансового факультета
Ми	нского инс	титута управления
		С. А. Медведев
"	,,	2005 p

РАБОЧАЯ ПРОГРАММА

по дисциплине

"Объектно-ориентированное проектирование и программирование" для студентов специальности 1—40 01 02-02 «Информационные системы и технологии (в экономике)» учётно-финансового факультета

Дневное отделение	Заочное отделение
Курс 2 Семестры 3, 4 Лекции —70(36/34) часов из них КСР 32(14/18) часов Лабораторные) занятия 90(36/54) часов из них КСР - часов Всего 265 часа из них КСР 30 часов Курсовая работа 4 семестр	Курсы 2,3 Семестры 3,4,5 Лекции 20(10/6/4) часов Лабораторные занятия 24(/12/12) часа Всего 44 часа Курсовая работа 5 семестр Самостоятельная работа 116 часов Зачёт 4 семестр Экзамен - 5 семестр
Самостоятельная работа 105 часа Зачёт 3 семестр Экзамен 4 семестр	

Рабочая программа составлена к.т.н., доцентом Демидовичем Е.М на основании типовой программы "Объектно-ориентированное программирование и проектирование" регистрационный №ТД-40-071 /тип, утвержденной УМО вузов Республики Беларусь по образованию в области информатики и радиоэлектроники от 29 октября 2004 г.

Рабочая программа обсуждена н онных систем(" "	иа заседании кафедры автоматизированных информаци- 2005 г., протокол №).
Зав. кафедрой АИС	В.И. Курмашев
	и одобрена научно-методической комиссией информа- х дисциплин Минского институт управления 2
Председатель комиссии	В.В. Гедранович

І. Цели и задачи дисциплины, её место в учебном процессе

1.1. Цель преподавания дисциплины

Цель изучения данной дисциплины – овладение знаниями и навыками проектирования информационных систем с использованием языка программирования С++, формирование компьютерной грамотности и подготовка студентов к использованию современных компьютеров и объектно-ориентированной технологии программирования в качестве инструмента для решения практических задач в своей предметной области.

1.2. Задачи изучения дисциплины

В результате освоения курса «Объектно-ориентированное программирование и проектирование» студент должен:

знать:

- основные теоретические понятия ООП, механизмы реализации объектно-ориентированного подхода, достоинства и недостатки объектной технологии программирования, тенденции и перспективы развития объектно-ориентированного подхода в программировании;

уметь характеризовать:

- выбор методов и средств объектно-ориентированного подхода для реализации программных проектов;

уметь анализировать:

- предметную область решаемых задач с целью использования объектно-ориентированного подхода для их реализации;

приобрести навыки:

- приобрести практические навыки разработки программ на языка С++;
- владения приемами объектно-ориентированного анализа предметной области и требований к разрабатываемым программам;
- самостоятельно проектировать информационные системы с использованием языка программирования C++;
- программирования объектов с использованием всех возможностей объектноориентированных технологий;
- приобрести навыки эффективной работы в визуальных средах.

1.3. Связь дисциплины с другими учебными дисциплинами

Изучение дисциплины основано на использовании знаний, полученных студентами по дисциплинам «Высшая математика» и «Основы информатики и программирования». Материал дисциплины необходим для изучения дисциплин визуальные средства разработки приложений, языки программирования для разработки сетевых приложений, а также при прохождении производственной (преддипломной) практики, в дипломном проектировании.

1.4 . Структура дисциплины

1.4.1. Программа учебной дисциплины рассчитана:

для дневной формы обучения

на 160 часов учебных занятий, из них 70 часов— лекции, 90 - лабораторные занятия; дисциплина изучается в 3-м и 4-м семестрах;

по дисциплине предусмотрена курсовая работа в 4-м семестре;

формами контроля по дисциплине являются зачёт в 3-м семестре и экзамен в 4-м семестре;

для заочной формы обучения

на 44 часа учебных занятий, из них 20 часов - лекции, 24 часа лабораторные занятия; дисциплина изучается в 3-м, 4-м и 5-м семестрах;

по дисциплине предусмотрена курсовая работа в 5-м семестре;

формами контроля по дисциплине являются зачет в 4 -м семестре, экзамен в 5-м семестре.

1.4.2 План-график дисциплины на семестр (дневное обучение)

Таблица 1.1

Семестр	Число	Число часов						
	недель	Общее число ча- Контролируемая самостоятельная раплану) бота			Самостоятельная работа			
		лк	ПЗ	лб	Лк	П3	лб	
3	18	36	-	36	14	-		45
4	18	34	-	54	18	-		60

1.4.3. План-график дисциплины на неделю (дневное обучение)

Таблица 1.2

Семестр	Всего учебных часов в неделю	Число часов				
		лекции	лабораторные раб.	Самостоятельная работа		
3	4	2	2	2,5		
4	5	2	3	3,2		

1.4.4. План-график дисциплины на семестр (заочное обучение)

Таблица 1.3

Семестр	Число недель зач		Число часов				
	экз. сессии	Лекции	Лабораторные работы	Контрольная работа	Самостоятельная работа		
3	3	10	-	-	30		
4	3	6	12		40		
5	3	4	12		46		

II. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Распределение часов по темам и видам занятий

№	Наименование	Объём в часах						
разде- ла	разделов (тем)	Дн	евное		Заочное отделение			
			дит.	Из них КСР		лк	лб	
		лк	лб	ЛК	лб			
1	2	3	4	5	6	7	8	
I.	Основные теоретические понятия ООП.	6	4	6		2	-	
1.1	Введение. Предмет курса, его задачи, структура Возникновение ООП. Понятие объекта и фундаментальные характеристики ООП (инкапсуляция, наследование, полиморфизм). Использование спецификатора const.					0,5		
1.2	Язык программирования С++ как эталонный пример использования ООП. Отличие языка С++ от процедурных языков программирования.	2		2		0,5		
1.2.1	Перегрузка функций. Значение формальных параметров по умолчанию в языке C++.	2		2		0,5		
1.2.2	Указатели и ссылки. Указатель на тип void. Указатель this. Использование операторов new и delete.	2	4	2		0,5		
II.	Понятие объекта.	8	12	2		2	-	
2.1	Понятие классов и экземпляров классов. Описание классов(class, struct, union). Вложенные классы. Определение объектов при помощи классов. Конструкторы и деструкторы. Атрибуты доступа к компонентам классов. Объявление и определение методов класса. Вызов членов класса. Конструкторы и деструкторы.	4	8	2		0,5		
2.2	Перегрузка конструкторов. Конструкторы копий.	2	4			1		
2.3	Механизмы реализации объектов на уровне объектного кода.	2				0,5		
III.	Перегрузка операторов.	8	8	2		1	-	
3.1	Основные положения перегрузки операторов.	2		2				
3.2	Перегрузка унарных операторов.	2	4			0,5		
3.3	Перегрузка бинарных операторов.	2	4					
3.4	Перегрузка операторов отношения и логических операторов. Функция оператор как член класса и как friend функция. Ограничения на перегрузку операций.	2				0,5		

IV.	Механизмы взаимодействия объектов.	14	12	4	3	
4.1.	Дружественные функции и классы.	2	4			
	Доступ к компонентам классов в зави-					
	симости от атрибутов доступа.					
4.2.	Введение в наследование. Базовые и	2			1	
	производные классы. Ограничение досту-					
	па. Наследование свойств и модификаторы					
	доступа. Одиночное наследование. Дос-					
	туп к компонентам производных и ба-					
	зовых классов.					
4.3.	Множественное наследование. Пере-	4	4	2	1	
	определение членов базового класса в					
	производном. Конструкторы базовых и					
	производных классов. Механизмы вызо-					
	вов конструкторов и деструкторов при					
	множественном наследовании.					
4.4.	Виртуальные базовые классы. Чисто	2			1	
	виртуальные функции. Абстрактные					
	базовые классы и их использование для					
	описания объектной модели.		<u> </u>			
4.5.	Использование указателей на базовые	2	4			
	классы и производные классы. Вирту-					
1.6	альные методы.	1 2	+	1		
4.6.	Реализация механизмов раннего и позд-	2		2		
	него связывания на примерах объект-					
	ных кодов.					
	пыл кодов.					
Итого	Дневное обучение (3-й семестр)	36	36	14		
Итого V.		36	36 12	14	2	
-	Дневное обучение (3-й семестр)				2	
V.	Дневное обучение (3-й семестр) Организация ввода/вывода.	8	12	4		
V.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока.	8	12	4		
V.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и меха-	8	12	4		
V. 5.1.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков.	8 4	12	2		
V.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизартные клас	8	12	4		
V. 5.1.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода.	8 4	4	2	1	
V. 5.1.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации	8 4	4	2		
V. 5.1.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов.	8 4	4	2	1	
V. 5.1.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа.	8 4	4	2	1	
5.1. 5.2. 5.3.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа.	8 4	4	2	1	
V. 5.1. 5.2. 5.3.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Заочное обучение(3-й семестр)	2 2	4 4	2	1 10	
V. 5.1. 5.2. 5.3. Итого VI.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Заочное обучение(3-й семестр) Шаблоны.	2 2 6	12 4 4 4	2	1 1 10 4	8
V. 5.1. 5.2. 5.3.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Заочное обучение(3-й семестр) Шаблоны. Понятие шаблона. Шаблоны функций и	2 2	4 4	2	1 10	8 4
V. 5.1. 5.2. 5.3. Итого VI.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Заочное обучение(3-й семестр) Шаблоны. Понятие шаблона. Шаблоны функций и классов. Шаблоны и наследование.	2 2 6	12 4 4 4	2	1 1 10 4	
V. 5.1. 5.2. 5.3. Итого VI.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Заочное обучение(3-й семестр) Шаблоны. Понятие шаблона. Шаблоны функций и классов. Шаблоны и наследование. Шаблоны и друзья. Шаблоны и стати-	2 2 6	12 4 4 4	2	1 1 10 4	
V. 5.1. 5.2. 5.3. Итого VI. 6.1.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Заочное обучение(3-й семестр) Шаблоны. Понятие шаблона. Шаблоны функций и классов. Шаблоны и наследование. Шаблоны и друзья. Шаблоны и статические члены.	8 4 2 2 6 2	12 4 4 4 12 4	2	1 1 10 4 2	4
V. 5.1. 5.2. 5.3. Итого VI.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Заочное обучение(3-й семестр) Шаблоны. Понятие шаблона. Шаблоны функций и классов. Шаблоны и наследование. Шаблоны и друзья. Шаблоны и статические члены. Введение в стандартную библиотеку	2 2 6	12 4 4 4	2	1 1 10 4	
V. 5.1. 5.2. 5.3. Итого VI. 6.1.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Заочное обучение(3-й семестр) Шаблоны. Понятие шаблона. Шаблоны функций и классов. Шаблоны и наследование. Шаблоны и друзья. Шаблоны и статические члены.	8 4 2 2 6 2	12 4 4 4 12 4	2	1 1 10 4 2	4
V. 5.1. 5.2. 5.3. Итого VI. 6.1.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Заочное обучение(3-й семестр) Шаблоны. Понятие шаблона. Шаблоны функций и классов. Шаблоны и наследование. Шаблоны и друзья. Шаблоны и статические члены. Введение в стандартную библиотеку шаблонов(STL). Контейнеры. Итераторы.	8 4 2 2 6 2	12 4 4 4 12 4	2 2	1 1 1 1 1 1 2 2 0,5	4
V. 5.1. 5.2. 5.3. Итого VI. 6.1.	Дневное обучение (3-й семестр) Организация ввода/вывода. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Понятие манипулятора. Реализации пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Заочное обучение(3-й семестр) Шаблоны. Понятие шаблона. Шаблоны функций и классов. Шаблоны и наследование. Шаблоны и друзья. Шаблоны и статические члены. Введение в стандартную библиотеку шаблонов(STL). Контейнеры. Итерато-	8 4 2 2 6 2	12 4 4 4 12 4	2	1 1 10 4 2	4

	ровки. Математические алгоритмы.					
VII.	Исключения.	8	8	4	2	4
7.1.	Основы обработки исключений в С++. Генерация исключений. Перехватывание исключений. Повторная генерация исключения.	4	4	2	1	2
7.2.	Обработка неожидаемых исключений Исключения и наследование. Иерархия исключений стандартной библиотеки.	4	4	2	1	2
Итого	Заочное обучение(4-й семестр)				6	12
VIII.	Списки.	6	10	6	1,5	6
8.1.	Очереди, стеки, кольца, деревья. Основные понятия и определения.	2	4	2	1	2
8.2.	Организация, структурные элементы и создание объектно-ориентированных списков.	4	6	4	0,5	4
IX.	Объектно-ориентированное программирование в визуальных средах.	4	12	2	2,5	6
9.1.	Изучение основных классов используемых при разработке программ. Классы элементов управления. Работа с текстовыми документами. Панели инструментов и строка состояния. Печать документов и организация работы в визуальной среде.	4	12	2	2	4
X.	Обзор современных инструментальных средств с поддержкой объектноориентированных технологий.	2			0,5	2
Итого	Дневное обучение (4-й семестр)	34	54	18		1.0
	Заочное обучение (5-й семестр) Итого по дисциплине	70	90	32	20	12 24
	итого по дисциплине	70	70	34	20	47

2.2. Наименование тем лекций и их содержание

Раздел 1. Основные теоретические понятия ООП.

Тема 1.1. Введение. Предмет курса, его задачи, структура. Возникновение ООП. Понятие объекта и фундаментальные характеристики ООП (инкапсуляция, наследование, полиморфизм). Использование спецификатора const.

Л.1, с. 7-16; Л.2, с. 57-76; Л.3, с. 9-18;

- **Тема 1.2.** Язык программирования C++ как эталонный пример использования ООП. Отличие языка C++ от процедурных языков программирования.
- 1.2.1. Перегрузка функций. Значение формальных параметров по умолчанию в языке C++. Л.1, с. 227- 229; Л.2, с. 191-197; Л.3, с. 36-41;
- 1.2.2. Указатели и ссылки. Указатель на тип void. Указатель this. Использование операторов new и delete.
 - Л.1, с. 154-1565, с. 215-226; Л.2, с. 136-140; Л.3, с. 113-118, 127-143;

Раздел 2. Понятие объекта.

Тема 2.1. Понятие классов и экземпляров классов. Описание классов(class, struct, union). Вложенные классы. Определение объектов при помощи классов. Конструкторы и деструкторы. Атрибуты доступа к компонентам классов. Объявление и определение методов класса. Вызов членов класса. Конструкторы и деструкторы.

Л.1, с.281-294; Л.2, с. 270-292; Л.3, с. 43-72;

Тема 2.2. Перегрузка конструкторов. Конструкторы копий.

Л.1, с. 287-295; Л.2, с. 292-297; Л.3, с. 143-157;

Тема 2.3. Механизмы реализации объектов на уровне объектного кода.

Л.2, с. 242-253; Л.3, с. 401-411;

Раздел 3. Перегрузка операторов.

Тема 3.1. Основные положения перегрузки операторов.

Л.1, с. 322-323; Л.2, с. 309-312; Л.3, с. 175-176.

Тема 3.2. Перегрузка унарных операторов.

Л.1, с. 323-325; Л.2, с. 322-323; Л.3, с. 186-189.

Тема 3.3. Перегрузка бинарных операторов.

Л.1, с. 325-335; Л.2, с. 311-315; Л.3, с. 178-185.

Тема 3.4. Перегрузка операторов отношения и логических операторов.

Функция оператор как член класса и как friend функция. Ограничения на перегрузку операций. Л.1, с. 315-321; Л.2, с. 323-334; Л.3, с. 185-186.

Раздел 4. Механизмы взаимодействия объектов.

- **Тема 4.1.** Дружественные функции и классы. Доступ к компонентам классов в зависимости от атрибутов доступа. Л.1, с. 295-302; Л.2, с. 326-330; Л.3, с. 97-107.
- **Тема 4.2.** Введение в наследование. Базовые и производные классы. Ограничение доступа. Наследование свойств и модификаторы доступа. Одиночное наследование. Доступ к компонентам производных и базовых классов. Л.1, с. 336-348; Л.2, с. 349-357; Л.3, с. 205-222.
- **Тема 4.3.** Множественное наследование. Переопределение членов базового класса в производном классе. Конструкторы базовых и производных классов. Механизмы вызовов конструкторов и деструкторов при множественном наследовании.

Л.1, с. 349-358; Л.2, с. 443-462; Л.3, с. 223-229.

Тема 4.4. Виртуальные базовые классы. Чисто виртуальные функции. Абстрактные базовые классы и их использование для описания объектной модели.

Л.1, с. 359-369; Л.2, с. 358-372, с. 450-456; Л.3, с. 229-239.

Тема 4.5. Использование указателей на базовые классы и производные классы. Виртуальные методы. Л.1, с. 369-375; Л.2, с. 474-479; Л.3, с. 303-317.

Тема 4.6. Реализация механизмов раннего и позднего связывания на примерах объектных кодов. Л.2, с. 462-473; Л.3, с. 357-376.

Раздел 5. Организация ввода/вывода.

Тема 5.1. Потоки ввода-вывода. Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода.

Л.1, с. 379-384; Л.2, с. 671-697; Л.3, с. 239-254;

Тема 5.2. Стандартные классы, объекты и механизмы файлового ввода/вывода.

Л.1, с. 385-408; Л.2, с. 703-716.

Тема 5.3. Понятие манипулятора. Реализация пользовательских манипуляторов. Файлы последовательного доступа. Файлы произвольного доступа. Л.1, с. 409-444; Л.2, с. 698-707; Л.3, с. 254-302.

Раздел 6. Шаблоны.

Тема 6.1. Понятие шаблона. Шаблоны функций и классов. Шаблоны и наследование. Шаблоны и друзья. Шаблоны и статические члены.

Л.1, с. 230-236; Л.2, с. 377-392; Л.3, с. 325-337.

Тема 6.2. Введение в стандартную библиотеку шаблонов(STL). Контейнеры. Итераторы. Л.2, с. 485-499; Л.3, с. 419- 424.

Тема 6.3. Адаптеры контейнеров. Алгоритмы. Основные алгоритмы поиска и сортировки. Математические алгоритмы. Л.2, с. 592-606; Л.3, с. 453-462.

Раздел 7. Исключения.

Тема 7.1. Основы обработки исключений в C++. Генерация исключений. Перехватывание исключений. Повторная генерация исключения.

Л.1, с. 445-465; Л.2, с. 407-427; Л.3, с. 337-351.

Тема 7.2. Обработка неожидаемых исключений. Исключения и наследование. Иерархия исключений стандартной библиотеки.

Л.1, с. 465-485; Л.2, с. 429-441; Л.3, с. 351-357.

Раздел 8. Списки.

Тема 8.1. Очереди, стеки, кольца, деревья. Основные понятия и определения. Л.1, с. 370-373; Л.2, с. 519-533; Л.3, с. 435-446.

Тема 8.2. Организация, структурные элементы и создание объектно-ориентированных списков.

Л.2, с. 533-550; Л.3, с. 446-453.

Раздел 9. Объектно-ориентированное программирование в визуальных средах.

Тема 9.1. Изучение основных классов используемых при разработке программ. Классы элементов управления. Работа с текстовыми документами. Панели инструментов и строка состояния. Печать документов и организация работы в визуальной среде.

Л.2, с. 848-865;

Раздел 10. Обзор современных инструментальных средств, с поддержкой объектно-ориентированных технологий.

2.3 Лабораторные занятия, их наименование и объём в часах

	Наименование тем	Цель занятия		Объём і	в часах
№	лабораторных заня-		Днев	Дневное	
	тий		отдел	ение	ное
п.п			Ауди-	Из	отде-
			торн.	них	ление
			часы	КСР	
1	2	3	4	5	6
1	Объектно-	Разработка проектируемых	2		
	ориентированный	классов и функций.			
	анализ разрабаты-				
	ваемой программы.				
	Разработка описа-				
	ний базовых объек-				
	тов и их взаимодей-				
	ствий.				
2	Указатели и ссыл-	Разработка программы с ис-	2		
	ки. Указатель на	пользованием ссылок.			
	тип void.				

Объекты и классы. Разработать программу с использованием: классов (объектов), внешнего доступа к компонентам объекта (friend), вложенных классов (объектов), и перегрузки конструкторы копий ектов), и перегрузки конструкторы. Составить программу перегрузка бинарных операторов Грузки операторов Г	_	0.4			1
ектов), внешнего доступа к компонентам объекта (friend), вложенных классов. 4 Перегрузка конетрукторы коний. Стов), и перегрузки конетрукторы коний. Стов), и перегрузки конетрукторы коний. Стов), и перегрузки конетрукторов. 6 Перегрузка унарных операторов. Тузки операторов. Тузки операторов +,-, ++, 7 Перегрузка бинарных операторов +,-, ++, 8 Дружественные функций и классы. Доступ к компонентам классов в зависимости от атрибутов доступа. Доступ к компонентам классов в зависимости от атрибутов доступа. В наследование (множественное), виртузальные мественное), виртузальные мественное), виртузальные методы. То наследованием множественного наследованием множественного наследованием множественного наследованием указателей и базовые классы в производные классы. Виртуальные методы. В пользованием указателей и механизмы консольного ввода/вывода. Разработать программу организации работы с потоками ввода вывода. Классы потоков. Стапдартные классы, объекты и механизмы файлового ввода/вывода. 12 Стандартные классьсь, объекты и механизмы файлового ввода/вывода. Разработать программу обработки файлов. В разработать программу с разработки файлов. В разработать программу с реализации пользовательских манипуляторов. В пользованием шаблонов и наследованием пользованием шаблонов и наследованием пользованием шаблонов и наследованием и пользованием шаблонов и наследованием пользованием шаблонов (классов, функций).	3			8	
Перегрузка конструкторов. Конструкторов. Конструкторов. Конструкторов. Конструкторов. Конструкторов. Конструкторов. Составить программу с использованием: классов (объектов), и перегрузка конструкторов. Составить программу перетружка операторов. Перегрузка бинаршых операторы перетрузка бинаршых операторы перегрузка конструкторы. Перегрузка конструкторы перегрузка конструкторы. Перегрузка конструкторы перегрузка конструкторы. Перегрузка конструкторы перегрузка конструкторы. Перегрузка конструкторы перегрузка конструкторы. Перегрузка конструкторы. Перегрузка конструкторы. Перегрузка конструкторы перегрузка конструкторы. Перегрузка конструкторы перегрузка конструкторы. Перегрузка констру		Инкапсуляция.			
ВВЛОЖЕШНЫХ КЛЯССОВ. Разработать программу с истовуюторых конструкторых конструкторых конструкторых конструкторых конструкторых конструкторов. Составить программу перепрузка бипарных операторов. Составить программу перепрузка бипарных операторов. Составить программу перепрузка операторов Составить программу перепрузка операторов. Составить программу перепрузка операторов. Составить программу перепрузка операторов. Составить программу перепрузка операторов. Составить программу с празваботать программу с празработать программу органые классы, объекты и механизмы консольного ввода/вывода. Разработать программу обработы с потоками ввода вывода. Составить программу обработки файлов. Составить программу с реализацией пользовательских манипуляторов. Составить программу с реализацией пользовательских манипуляторов. Составить программу с пользованием шаблонов (классов, функций). Разработать программу с пользованием шаблонов (классов, функций). Составить программу с пользовательства программу с пользовательства программу с пользовательства программу с пользовательства про					
4 Перегрузка конструкторы. Конструкторы коний. Разработать программу с использованием: классов (объекты и механизмы файлового ввода Вывода. 4 4 6 Перегрузка унарных операторов. Составить программу перетрузки операторов +, - + +, - 4 7 Перегрузка бинарных операторов. Составить программу перетрузки операторов =, <, >, +, - и др. 4 8 Дружественные функции и классы. Доступ к компонентам классов в завысимости от атрибутов доступа. Разработать программу с реализацией дружественных функций и классов. 4 2 9 Наследование (мпожественное), виртуальные классы и пользованием множественного наследования. Разработать программу с использованием множественных функций и классов. 4 2 10 Использование указателей и классы и потоков. Стандартные классы, объекты и механизмы консольного ввода вывода. Разработать программу организации работы с потоками ввода вывода. 4 2 11 Потоки ввода вывода. Разработать программу объекты и механизмы файлового ввода вывода. Разработать программу объекты и механизмы файлового ввода вывода. 4 2 12 Стандартные классы, объекты и механизмы файлового ввода вывода. Разработать программу с реализацией пользовательских манинуляторов. 4 2 13 Понятие манинуляторов.			. , , , , , , , , , , , , , , , , , , ,		
рукторов. Конструкторы копий. 6 Перегрузка унарпых операторов. 7 Перегрузка бинарных операторов. 8 Дружсственные функции и классы. Доступ к компонентам классов в зависимости от атрибутов доступа. 9 Наследование (множественное), виртуальное паследование уканые классы. Виртуальные классы. Виртуальные классы. Виртуальные классы. Виртуальные классы. Виртуальные классы. Виртуальные классы производные классы производные классы, объекты и механизмы консольного ввода/вывода. 11 Потокв ввода-вывода. Классы производные классы производные классы, объекты и механизмы консольного ввода/вывода. 12 Стандартные классы, объекты и механизмы файлового ввода/вывода. 13 Понятие манипуляторов. 14 Шаблоны функций и классов (объектов и пользованием множественного видет и дружной и механизмы консольного ввода/вывода. 14 Паблоны функций и классов (объектов и пользованием программу с использовательских манинуляторов. 15 Разработать программу ображной пользовательских манинуляторов. 16 Разработать программу ображной пользовательских манинуляторов. 17 Разработать программу с использовательских манинуляторов. 18 Паблоны функций и классов (объекты и механизмы пользователь программу с использователь программу с использователь программу с использователь программу с использователь (ких манинуляторов.)			вложенных классов.		
рукторы копий. Перегрузка унарных операторов. Перегрузка бинарных операторов. Перегрузка бинарных операторов. Перегрузка бинарных операторов. Перегрузка бинарных операторов трузки операторов тружки операторов трузки операторов трузки операторов трузки операт	4	Перегрузка конст-	Разработать программу с ис-	4	
рукторы копий. Перегрузка унарных операторов. Перегрузка бинарных операторов. Перегрузка бинарных операторов. Перегрузка бинарных операторов. Перегрузка бинарных операторов — Составить программу перегрузки операторов — Составить программу с использование мнале программу организации работать программу обработки файлов. Поизтие манинуляторов. Разработать программу обработки файлов. Разработать программу с использовательских манипуляторов. Разработать программу с использовательских манипуляторов. Разработать программу с использовательских манипуляторов. Разработать программу обработки файлов. Разработать программу обработки пользовательских манипуляторов.					
Перегрузка унарных операторов. Перегрузка бинарных операторов. Перетрузка бинарных операторов. Перетрузка бинарных операторов. Перетрузка бинарных операторов. Перетрузка операторов П			ектов), и перегрузки конст-		
6 Перегрузка унарных операторов. Составить программу перегрузки операторов +, -, +, 4 7 Перегрузка бинарных операторов. Составить программу перегрузки операторов =, <, >, +, - и др. 4 8 Дружественные функции и классы. Доступ к компонентам классов в зависимости от атрибутов доступа. Разработать программу с использование (множественное), виртуальное наследование им классов. 4 2 9 Наследование (множественное), виртуальные классы, и производные классы и пользование множественного наследования. Разработать программу с использованием множественного наследования. 4 2 10 Использование указателей и чисто виртуальных функций. Разработать программу органые классы, объекты и механизмы кольсовыного виртуальных функций и классы, объекты и механизмы кольсовыного ввода вывода. Разработать программу организации работы с потоками ввода вывода. 4 12 Стапдартные классы, объекты и механизмы файлового ввода/ вывода. Разработать программу обработки файлов. 4 2 13 Понятие манипуляторов. Разработать программу с использовательских манипуляторов. 4 2 14 Паблоны функций и классов, Шаблоны и наследование. Разработать программу с использованием шаблонов (классов, функций). 4 2					
операторов. прузки операторов +, -, + +, Перегрузка бинар- пых операторов. прузки операторов =, <, >, +, - и др. В Дружественные функции и классы. Доступ к компонентам классов в зависимости от атрибутов доступа. 9 Наследование (множественное), виртуальное наследование классов. 10 Использование указателей и базовые классы и производные классы. Виртуальные методы. 11 Потоки вводавые объекты и механизмы консольного ввода/ вывода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/ вывода. 12 Стандартные классы, объекты и механизмы файлового ввода/ вывода. 13 Понятие манипуляторов. 14 Шаблоны функций и классов, функций. 15 Полоки ввода вывода. Разработать программу обработать про	6	Перегрузка унарных		4	
7 Перегрузка бинарных операторов. Составить программу перегрузки операторов ==,<,>,+,- и др. 4 8 Дружественные функции и классы. Доступ к компонентам классов в зависимости от атрибутов доступа. Разработать программу с реализацией дружественных функций и классов. 4 2 9 Наследование (множественное), виртуальное наследование классов. Разработать программу с использование множественного наследования. 4 2 10 Использование указателей и бистов виртуальные классы и производные классы, объекты и механизмы консольного ввода-вывода. Разработать программу организации работы с потоками ввода вывода. 4 2 11 Потоки ввода-вывода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Разработать программу обработки файлов. 4 2 12 Стандартные классы, объекты и механизмы файлового ввода/ вывода. Разработать программу обработки файлов. 4 2 13 Понятие манипуляторов. Разработать программу с использовательских манипуляторов. 4 2 14 Шаблоны функций и классов, функций). Разработать программу с использованием шаблонов (классов, функций). 4 2					
В Дружественные функции и классы. Доступ к компонентам классов в зависимости от атрибутов доступа. Разработать программу с использование (множественное), виртуальное наследование классов. Разработать программу с использование указателей на базовые классы и производные классы. Виртуальные методы. Разработать программу с использование указателей и базовые классы и производные классы. Виртуальные методы. Разработать программы с использование методы. Разработать программы с использованием указателей и чисто виртуальных функций. Разработать программу организации работы с потоками ввода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Разработать программу обработки файлов. Разработать программу с реализацией пользовательских манипуляторов. Разработать программу с использовательских манипуляторов. Разработать программу с использовательских манипуляторов. Разработать программу с использовательских манипуляторов. Разработать программу с использованием шаблонов (классов, функций). Разработать программу с использованием шаблонов (классов, функций).	7			4	
1			1 1 2 1		
8 Дружественные функции и классы. Доступ к компонентам классов в зависимости от атрибутов доступа. Разработать программу с использование (множественное), виртуальное наследование классов. 4 2 10 Использование указателей и базовые классы и производные классы Виртуальные методы. Разработать программу с использованием множественного наследования. 4 2 11 Потоки вводавывода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Разработать программу организации работы с потоками ввода вывода. 4 2 12 Стандартные классы потоков. Стандартные классы, объекты и механизмы файлового ввода/ вывода. Разработать программу обработки файлов. 4 2 13 Понятие манипуляторов. Разработать программу с реализацией пользовательских манипуляторов. 4 2 14 Шаблоны функций и классов, Шаблоны и наследование. Разработать программу с использованием шаблонов и классов, Щаблоны и наследование. 4 2		india oneparapazi			
функции и классы. Доступ к компонентам классов в зависимости от атрибутов доступа. 9 Наследование (множественное), виртуальное наследование указателей на базовые классы и производные классы виртуальные классы. Виртуальные методы. 11 Потоки вводавывода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/ вывода. 12 Стандартные классьи, объекты и механизмы файлового ввода/ вывода. 13 Понятие манипуляторов. 14 Шаблоны функций и классов. Разработать программу с использованием указателей и чисто виртуальных функций. Разработать программу организации работы с потоками ввода вывода. Разработать программу обработки файлов. Разработать программу обработки файлов. Разработать программу обработки файлов. Разработать программу обработки файлов. Разработать программу с файлов. Разработать программу с файлов. Разработать программу с файлов.	8	Лружественные	*	4	2.
Доступ к компонентам классов в зависимости от атрибутов доступа. Разработать программу с использование множественного наследования. Разработать программу с использование множественного наследования. Разработать программы с использованием указателей и чисто виртуальных функций. Разработать программы с использованием указателей и чисто виртуальных функций. Разработать программу организации работы с потоками ввода вывода. Разработать программу обработы с потоками ввода вывода. Разработать программу обработы с потоками вода вывода. Разработать программу обработки файлов. Разработать программу обработки файлов. Разработать программу обработки файлов. Разработать программу с реализацией пользовательских манипуляторов. Разработать программу с использовательских манипуляторов. Разработать программу с использовательских манипуляторов. Разработать программу с использователь программу общения программу общ			-	'	-
там классов в зависимости от атрибутов доступа. 9 Наследование (миожественное), виртуальное наследование указателей на базовые классы и производные методы. 10 Использование указателей и базовые классы и производные методы. 11 Потоки вводавые оклассы, объекты и механизмы консольного ввода/вывода. 12 Стандартные классы, объекты и механизмы консольного ввода/вывода. 13 Понятие манипуляторов. 14 Шаблоны функций инаследование. 16 Имассов, Шаблоны и наследование. 17 Разработать программу обрабноство потоками ввода вывода. 18 Понятие манипуляторов. 19 Разработать программу обрабноство пользовательских манипуляторов. 20 Разработать программу обрабноство пользовательских манипуляторов. 21 Разработать программу обрабноство пользовательских манипуляторов. 22 Разработать программу обрабноство пользовательских манипуляторов. 23 Разработать программу обрабноство пользовательских манипуляторов. 24 Разработать программу с реализацией пользовательских манипуляторов. 25 Разработать программу с реализацией пользовательских манипуляторов. 26 Разработать программу с реализацией пользовательских манипуляторов. 27 Разработать программу с реализацией пользовательских манипуляторов.		1 - 7			
Симости от атрибутов доступа. Разработать программу с исмественное), виртуальное наследование указателей на базовые классы и производные классы. Виртуальные методы. Разработать программу органые классы. Виртуальные методы. Разработать программу организации работы с потоками ввода вывода. Разработать программу обработки файлов. Разработать программу с реализации пользовательских манипуляторов. Разработать программу с исми классов. Шаблоны и наследование. Разработать программу с исми классов, функций). Разработать программу с исми классов, функций. Разработать программу с исми классов, функций (классов, функций). Разработать программу с исми классов (классов, функций (классов, функций (классов) Разработать программу с исми классов (классов) Разработать		' ' =	функции и классов.		
Тов доступа. Разработать программу с истользованием множественного наследование (множественное), виртуальное наследование классов. Разработка программы с истользованием указателей и чисто виртуальных функций. Разработать программу организации работы с потоками ввода вывода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Разработать программу обработки файлов. Разработать программу с реализации пользовательских манипуляторов. Разработать программу с истора. Разработать программу с реализацией пользовательских манипуляторов. Разработать программу с истора. Разработать программу с истораму и и классов, функций).					
9 Наследование (множественное), виртуальное наследование множественное пользованием множественного наследования. 4 2 10 Использование указателей на базовые классы и производные классы. Виртуальные методы. Разработка программы с использованием указателей и чисто виртуальных функций. 4 2 11 Потоки вводавывода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Разработать программу организации работы с потоками ввода вывода. 4 12 Стандартные классы, объекты и механизмы файлового ввода/вывода. Разработать программу обработки файлов. 4 13 Понятие манипуляторов. манипуляторов. Разработать программу с реализацией пользовательских манипуляторов. 4 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с использованием шаблонов (классов, функций). 4					
жественное), вирту- альное наследова- ние классов. 10 Использование ука- зателей на базовые классы и производ- ные классы. Вир- туальные методы. 11 Потоки ввода- вывода. Классы по- токов. Стандартные классы, объекты и механизмы кон- сольного вво- да/вывода. 12 Стандартные клас- сы, объекты и меха- низмы файлового ввода/ вывода. 13 Понятие манипуляторов. 14 Паблоны функций и классов. Паблоны и наследования. 16 Использование ука- дагандар и пользованием множественно- го наследования. 16 Разработка программы с ис- пользованием указателей и чисто виртуальных функций. 16 Разработать программу орга- низации работы с потоками ввода вывода. 17 Ворганизмы файлового вода/ вывода. 18 Понятие манипуляторов. 19 Разработать программу с фреализацией пользователь- ских манипуляторов. 19 Разработать программу с ис- пользованием инаблонов (классов, функций).	0	-	Разработать программу с ис-	1	2
альное наследование классов. 10 Использование указателей на базовые классы и производные классы. Виртуальные методы. 11 Потоки вводавывода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. 12 Стандартные классы, объекты и механизмы файлового ввода/вывода. 13 Понятие манипуляторов. 14 Шаблоны функций и классов. Шаблоны и наследование.		`		-	2
Потоки вводавие классы потоков. Стандартные классы и механизмы консольного ввода/ вывода. Стандартные классы и механизмы файлового ввода/ вывода. Вода/ вывода. Понятие манипуляторов. Понятие манипуляторов. Понятие манипуляторов. Пользоватием шаблоны и наследование. Пользованием шаблонов (классов, функций).					
10 Использование указателей на базовые классы и производные классы и производные классы. Виртуальные методы. Разработка программы с использованием указателей и чисто виртуальных функций. 11 Потоки вводавывода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. Разработать программу организации работы с потоками ввода вывода. 4 12 Стандартные классы, объекты и механизмы файлового ввода/вывода. Разработать программу обработки файлов. 4 13 Понятие манипулятеров. Реализации пользовательских манипуляторов. Разработать программу с реализацией пользовательских манипуляторов. 4 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с использованием шаблонов (классов, функций). 4			то наследования.		
зателей на базовые классы и производные классы и производные классы. Виртуальные методы. 11 Потоки вводавывода. Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. 12 Стандартные классы, объекты и механизмы файлового ввода/ вывода. 13 Понятие манипуляторов. 14 Шаблоны функций и классов. Шаблоны и наследование. 15 Имбарт на базовые и чисто виртуальных функций. 16 Разработать программу организации работы с потоками ввода вывода. 17 Разработать программу обработки файлов. 18 Разработать программу обработки файлов. 19 Разработать программу с реализацией пользовательских манипуляторов. 19 Разработать программу с пользовательских манипуляторов. 10 Наблоны функций и наследование. 11 Классов, функций).	10		Danna Garres Imarma Augus a Ma	1	2
Классы и производ- ные классы. Вир- туальные методы. Разработать программу орга- вывода. Классы по- токов. Стандартные классы, объекты и механизмы кон- сольного вво- да/вывода. Разработать программу об- сы, объекты и меха- низмы файлового ввода/ вывода. Разработать программу об- работки файлов. Разработать программу с работки файлов. Разработать программу с реализацией пользователь- ских манипуляторов. Разработать программу с реализацией пользователь- ских манипуляторов. Разработать программу с реализацией пользователь- ских манипуляторов. Разработать программу с и классов. Шаблоны и наследование. Разработать программу с ис- пользованием шаблонов (классов, функций). Разработать программу с ис- пользованием шаблонов (классов, функций).	10			4	2
ные классы. Виртуальные методы. 11 Потоки вводавывода Классы потоков. Стандартные классы, объекты и механизмы консольного ввода/вывода. 12 Стандартные классы, объекты и механизмы файлового ввода/ вывода. 13 Понятие манипулятора. Разработать программу обвесты и механизмы файлового ввода/ вывода. 14 Паблоны функций и классов. Шаблоны и наследование. 15 Потоки ввода вывода. Разработать программу обранизмы файлового вода/ вывода. Разработать программу с реализацией пользовательских манипуляторов. Ских манипуляторов. Разработать программу с использователь (классов, функций).					
Туальные методы. 11 Потоки ввода- вывода Классы по- токов. Стандартные классы, объекты и механизмы кон- сольного вво- да/вывода. 12 Стандартные клас- сы, объекты и меха- низмы файлового ввода/ вывода. 13 Понятие манипуля- тора. Реализации пользовательских манипуляторов. 14 Шаблоны функций и классов. Шаблоны и наследование. Pазработать программу об- реализацией пользователь- ских манипуляторов Разработать программу с реализацией пользователь- ских манипуляторов Разработать программу с реализацией пользователь- ских манипуляторов Разработать программу с ис- пользованием шаблонов (классов, функций).		_	чисто виртуальных функции.		
11 Потоки ввода- вывода Классы по- токов. Стандартные классы, объекты и механизмы кон- сольного вво- да/вывода. Разработать программу орга- низации работы с потоками ввода вывода. 4 12 Стандартные клас- сы, объекты и меха- низмы файлового ввода/ вывода. Разработать программу об- работки файлов. 4 13 Понятие манипуля- тора. Реализации пользовательских манипуляторов. Разработать программу с реализацией пользователь- ских манипуляторов. 4 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с ис- пользованием шаблонов (классов, функций). 4		_			
вывода Классы потоков. Стандартные классы, объекты и механизмы консольного ввода. 12 Стандартные классы, объекты и механизмы файлового ввода/ вывода. 13 Понятие манипуляторов. 14 Шаблоны функций и наследование. 15 Вывода. Низации работы с потоками ввода вывода. 16 Ввода вывода. 17 Внаст на потоками ввода вывода. 18 Понятие манипуляторов. 19 Вывода вывода. 10 Ваблоны функций и наследование. 10 Вывода вывода. 11 Вывода вывода. 12 Стандартные классраю разработать программу обрасовательностих манипуляторов. 16 Виаблоны функций и наследование. 17 Вывода вывода. 18 Разработать программу с использовательного пользованием шаблонов (классов, функций).	11	i	D	1	
токов. Стандартные классы, объекты и механизмы консольного ввода. 12 Стандартные классы, объекты и механизмы файлового ввода/ вывода. 13 Понятие манипуляторам. 14 Шаблоны функций и наследование. 15 Ввода вывода. 16 Ввода вывода. 17 Разработать программу обрасний файлов. 18 Разработать программу с реализацией пользовательских манипуляторов. 19 Разработать программу с использователь пользователь пользователь пользованием шаблонов и классов, функций).	11			4	
классы, объекты и механизмы кон- сольного вво- да/вывода. 12 Стандартные клас- сы, объекты и механизмы файлового ввода/ вывода. 13 Понятие манипуляторамму с тора. Реализации пользовательских манипуляторов. 14 Шаблоны функций и классов. Шаблоны и наследование. 15 Классов, функций).			-		
механизмы кон- сольного вво- да/вывода. Разработать программу об- работки файлов. 4 2 12 Стандартные клас- сы, объекты и меха- низмы файлового ввода/ вывода. Разработать программу с реализацией пользователь- ских манипуляторов. 4 13 Понятие манипуля- тора. Реализации пользовательских манипуляторов. Разработать программу с ских манипуляторов. 4 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с пользованием шаблонов (классов, функций). 4 2		· · ·	ввода вывода.		
сольного вво-да/вывода. 12 Стандартные классы, объекты и механизмы файлового ввода/ вывода. Разработать программу обвода. 4 2 13 Понятие манипулятора. Реализации пользовательских манипуляторов. Разработать программу с исмих манипуляторов. 4 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с использованием шаблонов (классов, функций). 4 2		· ·			
да/вывода. Разработать программу об- сы, объекты и меха- низмы файлового ввода/ вывода. Разработать программу с реализацией пользователь- ских манипуляторов. 4 Тонятие манипуля- тора. Реализации пользовательских манипуляторов. Разработать программу с ских манипуляторов. 4 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с ис- пользованием шаблонов (классов, функций). 4 2					
12 Стандартные клас- сы, объекты и меха- низмы файлового ввода/ вывода. Разработать программу об- работки файлов. 4 2 13 Понятие манипуля- тора. Реализации пользовательских манипуляторов. Разработать программу с реализацией пользователь- ских манипуляторов. 4 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с ис- пользованием шаблонов (классов, функций). 4					
сы, объекты и механизмы файлового ввода/ вывода. 13 Понятие манипулятора. Разработать программу с реализацией пользовательских манипуляторов. Ских манипуляторов. 14 Шаблоны функций и наследование. Разработать программу с использованием шаблонов (классов, функций).	10		D ~	1	2
низмы файлового ввода/ вывода. 13 Понятие манипуля- тора. Реализации пользовательских манипуляторов. 14 Шаблоны функций и наследование. 16 Разработать программу с ис- пользовательских манипуляторов 2 Пользованием шаблонов (классов, функций).	12		-	4	2
ввода/ вывода. Понятие манипуля- тора. Реализации Разработать программу с до- реализацией пользователь- ских манипуляторов. 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с ис- пользованием шаблонов (классов, функций).			работки файлов.		
13 Понятие манипулятора. Реализации пользовательских пользовательских манипуляторов. Разработать программу с реализацией пользовательских манипуляторов. 4 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с использованием шаблонов (классов, функций). 4		-			
тора. Реализации реализацией пользователь- пользовательских ских манипуляторов 14 Шаблоны функций разработать программу с ис- и классов. Шаблоны и наследование. (классов, функций).					
пользовательских манипуляторов. ских манипуляторов. 2 14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с использованием шаблонов (классов, функций). 4 2	13			4	
манипуляторов. Разработать программу с ис- и классов. Шаблоны и наследование. 4 2 пользованием шаблонов (классов, функций). 2		_	1 -		
14 Шаблоны функций и классов. Шаблоны и наследование. Разработать программу с ис- пользованием шаблонов (классов, функций). 4 2			ских манипуляторов		
и классов. Шаблоны пользованием шаблонов (классов, функций).		· · · · · · · · · · · · · · · · · · ·			
и наследование. (классов, функций).	14		-	4	2
		и классов. Шаблоны			
Шаблоны и прузы		и наследование.	(классов, функций).		
шаолопы и друзья.		Шаблоны и друзья.			

1.5	Hamany non	Dana 6 amria	4	2
15	Использование	Разработка программы с ис-	4	2
	стандартной биб-	пользованием контейнерных		
	лиотеки шаблонов	классов и стандартных алго-		
	(STL). Контейнеры.	ритмов STL.		
1.0	Итераторы.	D. C		
16	Адаптеры контейне-	Разработка программы с ис-	4	
	ров. Основные ал-	пользованием контейнерных		
	горитмы поиска и	классов и		
	сортировки. Мате-	алгоритмы поиска и сорти-		
	матические алго-	ровки.		
15	ритмы.	0		
17	Генерация исклю-	Обработка исключений в	4	2
	чений. Перехваты-	программах.		
4.0	вание исключений.			
18	Обработка	Обработка в программах	4	2
	неожидаемых	неожидаемых исключений.		
	исключений.			
	Исключения и на-			
	следование.			
19	Очереди, стеки,	Разработка программы созда-	4	2
	кольца.	ния контейнерного класса		
		(одно или двух- направлен-		
		ный список).		
20	Организация,	Разработка программы созда-	6	4
	структурные эле-	ния контейнерного класса		
	менты и создание	бинарное дерево.		
	объектно-			
	ориентированных			
	списков.			
21	Разработка объектов	Разработка программы созда-	2	
	для реализации ин-	ния объектов для реализации		
	терфейса программы.	дружественного интерфейса.		
22	Вывод информации	Работа с текстом.	2	
	на экран.			
23	Панели инструмен-	Разработать программу до-	4	2
	тов и строка состоя-	бавления и удаления кнопок.		
	ния.			
24	Разработка объек-	Разработка программы	4	
	тов для работы с по-	для работы с потоками собы-		
	токами событий.	тий.		
	- CALLETT COURT INTE	<u> </u>	L	I

2.4 Курсовая работа, ее характеристика

Целью выполнения курсовой работы является закрепление и углубление теоретических знаний студентами по наиболее важным вопросам дисциплины, развития у них навыков объектно-ориентированного проектирования программного обеспечения, проверка способностей студентов квалифицированно применять полученные ими теоретические знания и практические навыки при решении практических задач.

Курсовая работа может выполняться по следующим основным направлениям:

создание пользовательского интерфейса для систем, работающих в текстовом и графическом режимах;

разработка графических и текстовых редакторов с минимальным набором типовых функциональных возможностей;

разработка диалоговых обучающих программ;

реализация игровых программ;

разработка библиотеки базовых классов.

2.5 Контролируемая самостоятельная работа (ООПП- ІІ курс)

Форма контроля:

• индивидуальные контрольные задания

Форма отчетности:

- ведомость с результатами.
 - 2.5.1 Контролируемая самостоятельная работа по темам лекций

Количество лекционных часов, запланированных на КСР - 32 часа.

Темы, выносимые на КСР в осеннем семестре(3 семестр-14 часов)

1. Отличие языка С++ от процедурных языков(2 часа, индивидуальное задание).

Язык программирования C++ как эталонный пример использования $OO\Pi$. Отличие языка C++ от процедурных языков программирования

Вопросы для самоконтроля

- 1. Перечислите фундаментальные характеристики ООП.
- 2. Что такое инкапсуляция?
- 3. Что такое наследование?
- 4. Что такое полиморфизм?

2. Перегрузка(2 часа, индивидуальное задание).

Перегрузка функций. Значение формальных параметров по умолчанию в языке C++. Л.1, с. 227- 229; Л.2, с. 191-197; Л.3, с. 36-41;

Вопросы для самоконтроля

- 1. Что такое перегрузка функций (операторов)?
- 2. Стандартная библиотека С++ содержит три функции:

double atof(const char *s);

int atoi(const char *s);

long atol(const char *s);

Эти функции возвращают численное значение, содержащееся в строке, на которую указывает s. Несмотря на то, что эти функции возвращают разные типы данных их нельзя перегрузить. Объясните почему?

3. Указатели и ссылки(2 часа, индивидуальное задание).

Указатель на тип void. Указатель this. Использование операторов new и delete.

Л.1, с. 154-1565,с. 215-226; Л.2, с. 136-140; Л.3, с. 113-118, 127-143;

Вопросы для самоконтроля

- 1. Что такое указатель?
- 2. Что такое ссылка?
- 3. В чем отличие указателя от ссылки?
- 4. Для чего используется указатель this?
- 5. Для каких целей в C++ включены операторы new и delete?

4. Классы(2 часа, индивидуальное задание)

Понятие классов и экземпляров классов. Описание классов(class, struct, union). Вло-

женные классы. Определение объектов при помощи классов. Конструкторы и деструкторы. Л.1, с.281-294; Л.2, с. 270-292; Л.3, с. 43-72;

Вопросы для самоконтроля

- 1. В чем отличие между классом и экземпляром класса?
- 2. Что такое конструктор? Что такое деструктор? Когда они вызываются?
- 3. В чем отличие между классом и структурой? В чем отличие между классом и объединением?
 - 4. Приведите пример вложенных классов.
 - 5. Сколько может быть конструкторов(деструкторов) в описании класса?
- 6. Пусть задан следующий класс, каковы имена его конструктора и деструктора?

5. Перегрузка операторов(2 часа, индивидуальное задание).

Основные положения перегрузки операторов.

```
Л.1, с. 322-323; Л.2, с. 309-312; Л.3, с. 175-176;
```

Вопросы для самоконтроля

- 1. Что такое перегрузка операторов?
- 2. Какие операторы перегружаются в С++?
- 3. В чем отличие перегрузки унарных и бинарных операторов?

6. Наследование(2 часа, индивидуальное задание)

Множественное наследование. Переопределение членов базового класса в производном классе. Л.1, с. 349-358; Л.2, с. 443-462; Л.3, с. 223-229;

Вопросы для самоконтроля

- 1. Для чего используется ключевое слово protected(public, privite)
- 2. Дан следующий фрагмент кода. В каком порядке вызываются конструкторы и деструкторы?
- class myclass: public A, public B, public C \{....\}
- 3. Когда базовый класс наследуется производным классом как открытый, что происходит с его открытыми членами? Что происходит с его закрытыми членами?
- 4. Когда базовый класс наследуется производным классом как закрытый, что происходит с его открытыми и открытыми членами?
- 5. При наследовании одного класса другим, когда вызываются конструкторы классов? Когда вызываются деструкторы классов?
 - 6. Объясните, зачем может понадобиться виртуальный базовый класс.

7. Механизмы связывания объектов(2 часа, индивидуальное задание).

Реализация механизмов раннего и позднего связывания на примерах объектных кодов. Л.2, с. 462-473; Л.3, с. 357-376;

Вопросы для самоконтроля

- 1. Что такое раннее связывание?
- 2. Что такое позднее связывание?
- 3. Как реализуется раннее связывание?
- 4. Как реализуется позднее связывание?

Темы, выносимые на КСР в весеннем семестре(4 семестр-18 часов)

8. Потоки ввода-вывода(2 часа, индивидуальное задание)

Понятие потока. Иерархия классов ввода-вывода. Основные уровни иерархии. Классы потоков. Л.1, с. 379-384; Л.2, с. 671-697; Л.3, с. 239-254;

Вопросы для самоконтроля

- 1. Что такое пользовательская функция вывода?
- 2. Что такое пользовательская функция ввода?
- 3. Что такое манипуляторы ввода/вывода?
- 4. Какие создаются встроенные потоки, когда начинается выполнение программы на C++?
- 5. Какой заголовок должен быть включен в программу для использования манипуляторов ввода/вывода с параметрами?
- 6. Напишите программу, которая бы устанавливала флаги для потока так, чтобы целые, если они положительные, выводились со знаком +.

9. Файловый ввод/вывод(2часа, индивидуальное задание).

Стандартные классы, объекты и механизмы файлового ввода/вывода.

Л.1, с. 385-408; Л.2, с. 703-716;

Вопросы для самоконтроля

- 1. Изобразите иерархию классов файлового ввода-вывода.
- 2. Какие классы используются для файлового ввода?
- 3. Какие классы используются для файлового вывода?
- 4. Какие имеются типы файлов?

10. Библиотека STL(2часа, индивидуальное задание).

Адаптеры контейнеров. Алгоритмы. Основные алгоритмы поиска и сортировки. Математические алгоритмы. Л.2, с. 592-606; Л.3, с. 453-462;

Вопросы для самоконтроля

- 1. Что такое адаптеры контейнеров?
- 2. Какие типы алгоритмов реализованы в библиотеке?
- 3. Перечислите основные алгоритмы поиска?

11. Исключения(2часа, индивидуальное задание)

Основы обработки исключений в C++. Генерация исключений. Перехватывание исключений. Л.1, с. 445-465; Л.2, с. 407-427; Л.3, с. 337-351;

Вопросы для самоконтроля

- 1. Что может произойти при возбуждении исключительной ситуации, для которой не задано соответствующей инструкции catch?
- 2. Какая инструкция catch перехватывает все типы исключительных ситуаций?
 - 3. Что неправильно в данном фрагменте программы?

```
try { // ...
throw 20;
}
catch (int * ip) {
//...
```

Предложите способ исправления предыдущего фрагмента.

- 4. Объясните, в чем разница между функционированием операторов new и new(nothrow), если при выделении памяти происходит ошибка.
- 5. Напишите обычные формы инструкций try, catch и throw. Опишите словами их функции.
- 6. Объясните, как совместная работа инструкций try, catch и throw обеспечивает в C++ обработку исключительных ситуаций.

12. Неожидаемые исключения (2часа, индивидуальное задание)

Обработка неожидаемых исключений. Исключения и наследование. Иерархия исключений стандартной библиотеки. Л.1, с. 465-485; Л.2, с. 429-441; Л.3, с. 351-357;

Вопросы для самоконтроля

- 1. Что такое неожидаемые исключения?
- 2. Какие исключения предусмотрены библиотекой С++?

13. Списки(2 часа, индивидуальное задание)

Очереди, стеки. Л.1, с. 370-373; Л.2, с. 519-533; Л.3, с. 435-446;

Вопросы для самоконтроля

- 1. Какие операторы используются для создания динамических списков?
- 2. Какие типы списков вы знаете?
- 3. Что такое очередь (стек)?
- 4. В чем отличие очереди от стека?
- 5. Что надо изменить в очереди, чтобы она стала стеком?

14. Объектно-ориентированные списки (4 часа, индивидуальное задание).

Организация, структурные элементы и создание объектно-ориентированных списков.

Л.2, с. 533-550; Л.3, с. 446-453;

Вопросы для самоконтроля

- 1. Что такое дерево?
- 2. Что такое корень дерева?
- 3. Что такое лист дерева?
- 2. Что такое бинарное дерево?

15. Работа в визуальной среде(2 часа, индивидуальное задание).

Изучение основных классов используемых при разработке программ. Классы элементов управления. Работа с текстовыми документами. Панели инструментов и строка состояния. Печать документов и организация работы в визуальной среде.

Л.2, с. 848-865;

Вопросы для самоконтроля

- 1. Что понимается под элементами управления?
- 2. Что такое строка состояния?
- 3. Что такое панель управления?

III. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

3.1. Основная литература:

- 1. Подбельский В.В. Язык C++. М.: Финансы и статистика, 2000. 560 с.
- 2. Страуструп Б. Язык программирования С++. М.:СПб.: «Издательство БИ-НОМ» - «Невский диалект», 2001 г. – 1099 с.
 - 3. Шилдт Г. Самоучитель С++. СПб.: BHV Санкт-Петербург, 1998. 688 с.

3.2. Дополнительная литература:

- 4. Архангельский А.Я. Программирование в C++Builder 4. М.: ЗАО Издательство БИ-НОМ", 1999.
- 5. Х.Дейтел, П.Дейтел. Как прграммировать на C++: Пер. с англ. М.: ЗАО «Издательство БИНОМ». 1999 г. 1024 с.: ил.
 - 6. Х.Дейтел, П.Дейтел. Как прграммировать на C++: Пер. с англ. М.: ЗАО «Издательство БИНОМ», 2001 г. 1152 с.: ил.

- 7. Кнут Д.Э. Искусство программирования, том 1. Основные алгоритмы, 3-е изд.: Пер. с англ. : Уч. Пос. М. : Издательский дом «Вильямс», 2000. 720 с. : ил.
- 8. Кнут Д.Э. Искусство программирования, том 2. Получисленные алгоритмы , 3-е изд.: Пер. с англ. : Уч. Пос. М. : Издательский дом «Вильямс», 2000. 832 с. : ил.
- 9. Кнут Д.Э. Искусство программирования, том 3. Сортировка и поиск, 2-е изд.: Пер. с англ.: Уч. Пос. М.: Издательский дом «Вильямс», 2000. 832 с.: ил.
- 10. Круглински Дэвид. Программирование на Microsoft Visual C++ 6.0 для профессионалов/Пер. с англ. СПб:Питер;М.: Издательско-торговый дом «Русская редакция», 2001. -864 с.: ил.
 - 11. Скляров В.А. Язык С++ и объектно-ориентированное программирование –Мн.: Выш. шк.1997.- 478 с.
- 12. Шилд Г. Программирование на Borland С++ для профессионалов. –Мн.: ООО "Попури", 1998.

IV. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ

Виды работы	
	часах
Изучение лекционного материала по конспекту лекций и реко-	0,5
мендуемой литературе(на один час лекций).	
Подготовка к лабораторным работам(на одну работу – 4 часа).	2
Самостоятельное изучение материала и ответов на вопросы.	10
Подготовка контрольной работы(для заочной формы обучения).	10
Написание отчета и подготовка к защите лабораторной работы	1
(на одну работу).	

V. ФОРМЫ КОНТРОЛЯ ЗА РАБОТОЙ СТУДЕНТОВ

- 1 Выборочный опрос на лекциях.
- 2. Проверка конспектов лекций студентов.
- 3. Проведение контрольных работ на потоке.
- 4. Подготовка к защите рефератов по заданным темам.
- 5. Опрос перед проведением лабораторных работ.
- 6. Собеседование при защите отчетов по лабораторным работам.

VI. ТЕХНИЧЕСКИЕ СРЕДСТВА, ПРИМЕНЯЕМЫЕ ДЛЯ ОБУЧЕНИЯ И КОНТРОЛЯ

Наименование	Форма использования	Ауд.	Кол-во
1. Компьютеры .	Лабораторные работы	28,40	13, 13

Доцент кафедры автоматизированных информационных систем, к.т.н.

Демидович Е.М.