

«

»

“ ”

“ ”
_____ .

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ Программирование

: 12.03.04

, :

: 2, : 3

,

		3
1	()	4
2		144
3	, .	85
4	, .	36
5	, .	0
6	, .	36
7	, .	36
8	, .	2
9	, .	11
10	, .	59
11	(, ,)	
12		

(): 12.03.04

216 12.03.2015 ., : 08.04.2015 .

: 1,

(): 12.03.04

, 2/1 20.06.2017

, 6 21.06.2017

:

, . .

:

,

:

. . .

1.

1.1

Компетенция ФГОС: ОПК.9 способность использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности; в части следующих результатов обучения:	
1.	,
1.	-
2.	
Компетенция ФГОС: ПК.16 способность разрабатывать инструкции для персонала по эксплуатации технического оборудования и программного обеспечения биомедицинских и экологических лабораторий; в части следующих результатов обучения:	
2.	
3.	

2.

2.1

(
---	--

.9. 1	
1. О принципах объектно-ориентированного подхода при конструировании программ на языках высокого уровня.	;
2. О принципах программирования "под WINDOWS" с использованием библиотеки MFC.	;
.9. 1	
3. Основы языков высокого уровня C и C++.	;
4. Синтаксис и способы программирования задач на указанных языках.	;
5. Систему классов библиотеки MFC.	;
.9. 2	
6. Принципы построения программ на базе MFC.	;
7. Конструировать программы с помощью языка C++ и библиотеки MFC, направленные на инженерные приложения.	;
8. Эффективно пользоваться электронным справочником MSDN.	;
.16. 2	
9. Читать, объяснять и корректировать тексты программ, направленных в среде C++.	;

.16. 3	
10.Самостоятельной разработки приложений под ОС WINDOWS" с использованием библиотеки MFC.	;

3.

3.1

	,	.	
: 3			
: - ().			
1. - (). , , .	0	1	1
: ++ .			
2. ++ : , , ; - ; .	0	1	3
: .			
3. - . - , , . : , , , .	0	1	4, 5
: .			
4. ().	0	1	4
: () .			
5. () . . : ; ; , .	0	2	3, 4
: - .			
6. - : , , , . , .	0	2	3, 4
: ().			
7. ().	0	2	3, 4
: .			
8.	0	2	3, 4
: .			
9. . . .	0	1	3, 4

:			
10.	.	0	1
	3, 4		
: - ++.			
11.	- ++.	0	1
	3, 4		
: std::string.			
12.	std::string.	0	1
	3, 4		
: .			
13.	try/catch.	0	1
	3, 4		
: .			
14.	.	0	1
	2, 5		
: STL			
15.	STL.	0	1
	2, 5		
: Windows			
16.	Windows.	0	2
Windows.	Windows-APL.		
Windows.	Windows.		
	5, 6		
Windows	Microsoft Foundation Classes (MFC).		
Windows	MFC.		
: .			
17.	.	0	2
	4, 5, 6		
: MFC			
18.	.	0	1
	4, 6		
: DLL			
19.	DLL.	0	2
. DLL	DLL.		
	DLL		
	DLL.		
	4, 5		
: .			

20.	.	0	2	7
:				
21.	. :	0	2	2, 6
: GDI-				
22.	(GDI).	0	2	5, 6
:				
23.	(MDI). , , MDI MFC. MDI Drag and Drop. MDI.	0	2	2, 5, 6
: App Wizard				
24.	. App Wizard.	0	2	6, 7

3.2

	,	.		
: 3				
: .				

3. () . , () - 4 :) ;) () ;) (x, y, v);)))) . .	4	4	7, 9	- App Wizard, Class Wizard ;
4. object, . - object. object, .	4	4	7, 9	- App Wizard, Class Wizard ;
: STL				
5. (STL), (. . " "). , . :) ;)) ;) .	2	2	7, 8, 9	- App Wizard, Class Wizard ;

<p>6. : (STL).</p> <p>" :</p> <p>" .</p> <p>" ,</p> <p>" sort STL.</p> <p>" -</p> <p>" Exception.</p> <p>" .</p> <p>1.</p> <p>(</p> <p>)</p> <p>().</p> <p>vector<AnsiString>.</p> <p>,</p> <p>.</p> <p>2.</p> <p>(</p> <p>)</p> <p>().</p> <p>deque<int>.</p>	2	2	7, 8, 9	-	App Wizard, Class Wizard ;
--	---	---	---------	---	----------------------------------

<p>7. : (STL). : " ." . , , STL " - Exception. " . 1. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; - 24 - AnsiString (). . 2. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; - .</p>	2	2	7, 8, 9	- App Wizard, Class Wizard ;
: Windows				

8. 1. " . " , " , 200 300 (,). " MForm, " . " . " TForm. " . 2. " . " , " . " - " , (OnCreate, Screen) " . " , " " " (OnMouseMove, Left, Top) 3. " , " ,				
: DLL				
9. " . " .	4	4	7, 8, 9	- App Wizard, Class Wizard
:				

<p>1. .</p> <p> . :</p> <p>1.</p> <p> .</p> <p>2.</p> <p> (</p> <p>()</p> <p>().</p> <p> .</p> <p> .</p> <p>:</p> <p>3.</p> <p> (</p> <p>()</p> <p>().</p> <p> .</p> <p> .</p> <p>4.</p> <p> (</p> <p>()</p> <p>().</p> <p> .</p>	4	4	7, 9	-	App Wizard, Class Wizard ;
---	---	---	------	---	----------------------------------

MFC.				
<p>" ;</p> <p>" App</p> <p>Wizard, Class Wizard ;</p> <p>" ,</p> <p>" ;</p> <p>" .</p> <p>1. ,</p> <p>2. - : - / . .</p> <p>[] : - / . .</p> <p>; . . . - . - , [2011]. - :</p> <p>http://courses.edu.nstu.ru/index.php?show=155&curs=351. - .</p>				
2		1, 10, 2, 3, 4, 5, 6, 7, 8, 9	26	0
<p>- :</p> <p>- ,</p> <p>- ,</p> <p>- ,</p> <p>- .</p> <p>: - . . - [] : , [2011]. -</p> <p>- / . . ; . . - . - : http://courses.edu.nstu.ru/index.php?show=155&curs=351. - .</p>				
3		1, 2, 3, 4, 5, 6	20	6
<p>: - . . - [] : , [2011]. -</p> <p>- / . . ; . . - . - : http://courses.edu.nstu.ru/index.php?show=155&curs=351. - .</p>				

5.

- , (. 5.1).

5.1

	-
	e-mail
	e-mail
	e-mail

6.

(),

-
15-

ECTS.

. 6.1.

: 3	
<i>Лабораторная:</i>	40
<i>Курсовая работа:</i>	20
<i>Экзамен:</i>	40

6.2

		/	/	
.9	1.	,	,	+
	1.	-		+
	2.	+	+	+
.16	2.	+	+	
	3.	+	+	

7.

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. Романов Е. Л. Си/Си ++. От дилетанта до профессионала [Электронный ресурс] : электронное учебное пособие : для 1-2 курсов направления 230100 "Информатика и вычислительная техника" / Романов Е. Л. - Новосибирск, 2010. - 1 электрон. опт. диск (CD-ROM). - Загл. с этикетки диска. - Рег. свидетельство №18891. - Режим доступа:http://elibrary.nstu.ru/source?bib_id=vtls000134024
3. Керниган Б. У. Язык программирования С : [пер. с англ.] / Брайан Керниган, Деннис Ритчи. - М. [и др.], 2009. - 289 с. : ил.
4. Подбельский В. В. Программирование на языке Си : учебное пособие для вузов по направлениям: "Прикладная математика и информатика", "Информатика и вычислительная техника", специальностям "Прикладная математика", "Вычислительные машины, комплексы, системы и сети управления" / В. В. Подбельский, С. С. Фомин. - М., 2007. - 600 с. : ил., табл.

1. Лафоре Р. Объектно-ориентированное программирование в С++ / Р. Лафоре ; [пер. с англ. А. Кузнецова, М. Назарова, В. Шрага]. - СПб., 2007. - 923 с. : ил.
2. Страуструп Б. Язык программирования С++. Специальное издание / Б. Страуструп ; пер. с англ. С. Анисимова и М. Кононова ; под. ред. Ф. Андреева и А. Ушакова. - М., 2005. - 1096 с. : ил.. - С авт. изм. и доп..

-

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.

8.1

1. Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: <http://courses.edu.nstu.ru/index.php?show=155&curs=351>. - Загл. с экрана.

8.2

1 Visual Studio

9.

-

1	(- , ,)	
2	(Internet)	

«

»

“ ”

“ ”
_____ .

УЧЕБНОЙ ДИСЦИПЛИНЫ
Программирование
: 12.03.04

		/	(, .)
().	.9	1.	
		1.	
STL.		1.	
		1.	
(MDI). MDI MFC. MDI MDI Drag and Drop. MDI.		1. 1. 2.	
		1. 2.	
		1.	

<code>()</code>	9	1.	-	
<code>:</code>		1.	-	
<code>DLL. DLL. DLL. DLL. DLL.</code>		1.	-	
<code>:</code>		1.	-	
<code>++</code>		1.	-	
<code>:</code>		1.	-	
<code>:</code>		1.	-	
<code>:</code>		1.	-	
<code>()</code>		1.	-	
<code>:</code>		1.	-	
<code>()</code>		1.	-	
<code>try/catch.</code>		1.	-	

std::string.	.9	1.	-	
- ++.		1.	-	
.		1.	2.	-
Windows. Windows. Windows. Windows-APL. Windows. . Windows Microsoft Foundation Classes (MFC). Windows MFC.		1.	2.	-
.		1.	2.	-
(GDI).		1.	2.	-
Wizard. App		2.		
.		2.		
.	.9 .17/	2. 3.	2.	

<div>1. " 200 300 (,). " MForm, " " " TForm. " 2. " . - (OnCreate, Screen) " " " " " (OnMouseMove, Left, Top) 3. " , .</div>	<div>.9 .17/</div>	<div>2. 3. 2.</div>	
--	--------------------	-----------------------------	--

<pre> - () , (4))));)) , (x, y, v);)))) , </pre>	<pre> .9 .17/ </pre>	<pre> 2. 2. </pre>	
--	----------------------	--------------------	--

<p> : (STL). : " " , sort STL. " - Exception. " . 1. ((). vector<AnsiString>. , . 2. ((). deque<int>. </p>	<p> .9 .17/ </p>	<p> 2. 2. </p>	
<p> object, - , object. object, . </p>		<p> 2. 2. </p>	

<div><div>: (STL). : " " " STL " - Exception. " 1. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; - 24 AnsiString (). 2. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; - .</div></div>	<div>.9 .17/</div>	<div>2. 2.</div>	
--	--------------------	----------------------	--

<p>1.</p> <p>2.</p> <p>()</p> <p>().</p> <p>3.</p> <p>()</p> <p>().</p> <p>4.</p> <p>()</p> <p>().</p>	<p>.9 .17/</p>	<p>2.</p> <p>2.</p>	
--	----------------	---------------------	--

<div>1. double</div> <div>2.</div> <div>2.</div>	<div>2.</div> <div>2.</div>		
<div>STL),</div> <div>2.</div> <div>2.</div>	<div>2.</div> <div>2.</div>		

	.9 .17/	2. 2.	
--	---------	----------	--

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.9 способность использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности	з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных	Классы окон. Дочерние окна. Элементы окон: полосы прокрутки, панель состояния, инструментальная панель. Назначение и состав библиотеки STL. Контейнеры, категории контейнеров, последовательные контейнеры. Итераторы. Объекты-функции. Векторы. Списки. Стеки. Очереди. Множество и мультимножество. Универсальные алгоритмы. Понятие многодокументного интерфейса (MDI). Рабочая область приложения, дочерние окна, просмотр и упорядочивание окон. Создание приложения MDI с помощью MFC. MDI и технологии Drag and Drop. Пример программного приложения MDI. Понятие объектно-ориентированного проектирования (ООП). Понятие объекта, объекта и класса, связь с переменной и типом данных. Шаблоны функций. Перегрузка и специализации шаблонов функций. Шаблоны функций сортировки. Шаблоны классов. Специализация шаблонов классов. Статические члены шаблонов классов. Класс - шаблон для вектора.		Экзамен
ОПК.9	у1. знать методы построения современных проблемно-ориентированных прикладных программных средств	Два примера разработки классов (дата и матрица переменной размерности). Действие и результат при выполнении элемента-функции или переопределения операции. Типы результата: пустой, базовый, указатель и ссылка на объект, объект. Временные объекты. Конструктор копирования и оператор присваивания, различия между ними. Друзья класса (функции и классы). Иерархия классов. Базовый и		Экзамен

		<p>производные классы. Доступность их элементов. Наследование. Синтаксис конструирования объектов производных классов. Краткий обзор особенностей C++ в сравнении с C: присваивание структур, структуры как формальные параметры и результаты функций, переопределение операций и функций; элементы-функции; операторы управления динамической памятью; ссылки. Множественное наследование. Виртуальные базовые классы. Абстрактные классы. Шаблоны. Назначение и состав библиотеки STL. Контейнеры, категории контейнеров, последовательные контейнеры. Итераторы. Объекты-функции. Векторы. Списки. Стеки. Очереди. Множество и мультимножество. Универсальные алгоритмы. Определение класса. Элементы-данные и элементы- функции класса. Общие, частные и защищенные элементы класса. Объекты и их свойства, время жизни и инициализация. Определение конструктора и деструктора, их свойства. Оператор присваивания. Типы объектов: внешний, автоматический, динамический, временный. Переопределение (перегрузка) операций. Пример. Переопределение функций. Пример. Особенности переопределения некоторых операций: преобразования типа; индексации; динамической памяти, присваивания. Полиморфизм и виртуальные функции. Абстрактные классы и чисто виртуальные функции. Виртуальные деструкторы. Понятие DLL. Создание DLL. Присоединение DLL к программе. DLL ресурсов. Пример создания DLL. Понятие графического интерфейса устройства (GDI). Контексты устройств. Графические классы. Графические объекты. Построение геометрических фигур. Создание типов кистей и перьев. Отображение теста. Растровые изображения. Создание шрифтов. Понятие</p>		
--	--	--	--	--

		<p>многодокументного интерфейса (MDI). Рабочая область приложения, дочерние окна, просмотр и упорядочивание окон.</p> <p>Создание приложения MDI с помощью MFC. MDI и технологии Drag and Drop.</p> <p>Пример программного приложения MDI. Понятие ресурсов. Типы ресурсов. Создание ресурсов. Редактирование ресурсов. Пример разработки ресурсов.</p> <p>Потоковый ввод-вывод в C++.</p> <p>Стандартный класс std::string. Конструкторы строк. Доступ к символам строк. Операции поиска. Операции вставки, замены и удаления символов строки. Операторы ввода-вывода строк. Таблица сообщений. Сообщения от клавиатуры. Сообщения от мыши. Управление исключениями. Понятие исключения. Обработка исключительных ситуаций. Генерирование исключений. Перехватывание исключений.</p> <p>Использование блоков try/catch. Неожиданные исключения и обработка завершения. Файлы программы для Windows. Типы данных в Windows. Обмен сообщениями между приложениями и ОС Windows. Понятие о Windows-APL.</p> <p>Структура приложения Windows. Определение класса окна. Регистрация класса окна. Создание и отображение окна. Цикл обработки сообщений. Пример простой Windows программы. Библиотека Microsoft Foundation Classes (MFC). Краткая характеристика библиотеки.</p> <p>Разработка приложений Windows с использованием MFC. Пример программы. Шаблоны функций. Перегрузка и специализации шаблонов функций. Шаблоны функций сортировки. Шаблоны классов. Специализация шаблонов классов. Статические члены шаблонов классов. Класс - шаблон для вектора.</p>		
ОПК.9	у2. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков	<p>Двоичные файлы произвольного доступа. Варианты в порядке увеличения сложности. На четверку: Вариант 1.</p> <p>Двоичный файл записей фиксированной длины.</p>	Курсовая работа Отчет по лабораторной работе	Экзамен

	<p>высокого уровня как средство программного моделирования изучаемых объектов и процессов</p>	<p>Написать функции записи в файл с заданным именем массива чисел типа double и чтения из файла числа с заданным номером.</p> <p>Программа должна работать следующим образом: если в командной строке один аргумент (это должно быть имя файла), то вызывается функция записи введенного с клавиатуры массива чисел, если аргументов два (имя файла и номер числа), то вызывается функция чтения из файла числа с заданным номером и вывод его на экран.</p> <p>На пятерку: Вариант 2.</p> <p>Открыть заданный текстовый файл и записать все его строки в двоичный файл записей переменной длины (каждая строка записывается в виде записи из двух полей [длина][содержимое]). После чего необходимо открыть созданный файл и вывести его содержимое на экран. Имя входного (текстового) файла задается в виде аргумента командной строки. Имя выходного (двоичного) файла генерируется программой: он должен иметь Классы окон. Дочерние окна. Элементы окон: полосы прокрутки, панель состояния, инструментальная панель. Понятие графического интерфейса устройства (GDI). Контексты устройств. Графические классы. Графические объекты. Построение геометрических фигур. Создание типов кистей и перьев. Отображение теста. Растровые изображения. Создание шрифтов. Понятие многодокументного интерфейса (MDI). Рабочая область приложения, дочерние окна, просмотр и упорядочивание окон. Создание приложения MDI с помощью MFC. MDI и технологии Drag and Drop. Пример программного приложения MDI. Понятие ресурсов. Типы ресурсов. Создание ресурсов. Редактирование ресурсов. Пример разработки ресурсов. Последовательные текстовые файлы. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Открыть заданный текстовый файл и вывести на экран строку с</p>		
--	---	---	--	--

		<p>заданным номером Имя файла и номер строки должны передаваться в программу в виде аргументов командной строки. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной не меняя порядка строк. Имена файлов задаются в виде аргументов командной строки. На пятерку: Вариант 3. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в обратном порядке строк. Имена файлов задаются в виде аргументов командной строки. Вариант 4. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в порядке убывания их длины. Имена файлов задаются в виде аргумен</p> <p>Разработать абстрактный базовый класс object, а для него предусмотреть чистые виртуальные методы. Разработанный ранее класс - ТД, сделать наследником от object. При этом переопределить чистые виртуальные методы класса object, наполнив их конкретным содержанием. Разработать класс - тип данных (ТД) с переопределением операторов. Разработать функции ввода, вывода, базовый набор содержательных операций над ТД (например, арифметических и сравнения) - 4 часа. Варианты типов данных: а) целые произвольной длины, представленные строкой цифр; б) то же, но во внутреннем двоичном представлении (динамический массив байтов); в) матрицы переменной размерности; г) разреженная матрица переменной размерности, ненулевые элементы представлены динамическим массивом с элементами (x, y, v); д) дроби; е) дата и время; ж) комплексные числа; з)</p>		
--	--	--	--	--

		<p>разрезены матрицы переменной размерности, ненулевые элементы представлены односвязным списком. Разработать шаблонный класс (с использованием стандартной библиотеки шаблонов STL), реализующей определенную структуру хранения данных (т.н. "контейнерный" класс). Данные в контейнере хранятся либо в виде указателей на объекты, либо в виде самих объектов. Структуры хранения данных: а) двоичное дерево; б) массив; в) односвязный линейный список; г) двусвязный циклический список. Разработать демонстрационную программу</p> <p>Создание и использование динамически присоединяемых библиотек. Создание консоли. Ввод и вывод в консоли. Обработка событий клавиатуры и мыши. Создание текстового редактора или калькулятора. Создание шаблона приложения. Создание однооконного приложения с помощью App Wizard. Создание многооконного приложения. Таблица сообщений. Сообщения от клавиатуры. Сообщения от мыши. Файлы программы для Windows. Типы данных в Windows. Обмен сообщениями между приложениями и ОС Windows. Понятие о Windows-APL. Структура приложения Windows. Определение класса окна. Регистрация класса окна. Создание и отображение окна. Цикл обработки сообщений. Пример простой Windows программы. Библиотека Microsoft Foundation Classes (MFC). Краткая характеристика библиотеки. Разработка приложений Windows с использованием MFC. Пример программы. Цель: освоить стандартную библиотеку шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, ее необходимо выполнить с помощью алгоритма sort из STL. " Обработка ошибок должна быть реализована с</p>		
--	--	--	--	--

		<p>помощью выбрасывания исключений - объектов класса <code>Exception</code>. " Работу программы проверить на файлах очень большого размера. Вариант 1. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Читать все строки входного файла в контейнер <code>vector<AnsiString></code>. Отсортировать строки в контейнере в алфавитном порядке, преобразовать к верхнему регистру и записать их в выходной файл. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Читать позиции начал строк входного файла в контейнер <code>deque<int></code>. Используя эти данные и реверсивный итератор записать в выходной файл строки. Цель: углубить знания стандартной библиотеки шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, поиска или перебора диапазона, эти действия необходимо выполнить с помощью алгоритмов библиотеки или встроенных в контейнеры функций STL " Обработка ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса <code>Exception</code>. " Работу программы проверить на файлах очень большого размера. Вариант 1. <code>typedef std::vector<AnsiString> Pages; std::list< Pages > Text;</code> Читать все строки заданного входного файла в структуру данных - список страниц. Страница - вектор из 24 строк, объектов класса <code>AnsiString</code> (последняя страница может быть неполной). Вывести на экран страницу с заданным номером. Вариант 2. <code>typedef std::vector<AnsiString> Pages; std::list< Pages > Text;</code> Читать все строки заданного входного файла в структуру данных - список страниц. С 1. Работа с инспектором объектов. " Создать приложение с главным окном,</p>		
--	--	--	--	--

		<p>появляющимся в центре экрана, с размерами 200 на 300 (высота, ширина). "</p> <p>Сменить название класса формы на MForm, сменить заголовок формы. "</p> <p>Запретить изменение размеров окна и распаивание во весь экран. "</p> <p>Экспериментальным путем выяснить за что отвечает каждое из свойств класс TForm. "</p> <p>Нарисовать собственную иконку используя редактор ресурсов и назначить ее приложению. 2. Обработка событий. "</p> <p>Доработать предыдущее приложение так, чтобы размеры и положение главной формы вычислялось автоматически при ее создании. Положение - правый нижний угол, высота и ширина равны четверти размеров экрана. (Событие OnCreate, объект Screen) "</p> <p>Поместить на форму стандартную кнопку. Сделать так, чтобы она "убегала" от курсора мыши. (Событие OnMouseMove, свойства Left, Top) 3. Диалоговая панель "</p> <p>Доработать приложение, поместив вторую кнопку и создав вторую форму. Сделать так, чтобы при нажатии на кнопку появлялась форма в модально</p>		
<p>ПК.16/ОУ</p> <p>способность разрабатывать инструкции для персонала по эксплуатации технического оборудования и программного обеспечения биомедицинских и экологических лабораторий</p>	<p>з2. уметь осваивать новые программные средства для профессиональной деятельности</p>	<p>Двоичные файлы произвольного доступа. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Двоичный файл записей фиксированной длины. Написать функции записи в файл с заданным именем массива чисел типа double и чтения из файла числа с заданным номером. Программа должна работать следующим образом: если в командной строке один аргумент (это должно быть имя файла), то вызывается функция записи введенного с клавиатуры массива чисел, если аргументов два (имя файла и номер числа), то вызывается функция чтения из файла числа с заданным номером и вывод его на экран. На пятерку: Вариант 2. Открыть заданный текстовый файл и записать все его строки в двоичный файл записей переменной длины (каждая строка записывается в виде</p>	<p>Курсовая работа</p> <p>Отчет по лабораторной работе</p>	

		<p>записи из двух полей [длина][содержимое]). После чего необходимо открыть созданный файл и вывести его содержимое на экран. Имя входного (текстового) файла задается в виде аргумента командной строки. Имя выходного (двоичного) файла генерируется программой: он должен иметь</p> <p>Последовательные текстовые файлы. Варианты в порядке увеличения сложности. На четверку: Вариант 1. Открыть заданный текстовый файл и вывести на экран строку с заданным номером Имя файла и номер строки должны передаваться в программу в виде аргументов командной строки. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной не меняя порядка строк. Имена файлов задаются в виде аргументов командной строки. На пятерку: Вариант 3. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в обратном порядке строк. Имена файлов задаются в виде аргументов командной строки. Вариант 4. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Скопировать строки из входного файла в выходной в порядке убывания их длины. Имена файлов задаются в виде аргумен</p> <p>Разработать абстрактный базовый класс object, а для него предусмотреть чистые виртуальные методы. Разработанный ранее класс - ТД, сделать наследником от object. При этом переопределить чистые виртуальные методы класса object, наполнив их конкретным содержанием. Разработать класс - тип данных (ТД) с переопределением операторов. Разработать функции ввода, вывода, базовый набор содержательных операций над ТД (например,</p>		
--	--	---	--	--

		<p>арифметических и сравнения) - 4 часа. Варианты типов данных: а) целые произвольной длины, представленные строкой цифр; б) то же, но во внутреннем двоичном представлении (динамический массив байтов); в) матрицы переменной размерности; г) разреженная матрица переменной размерности, ненулевые элементы представлены динамическим массивом с элементами (x, y, v); д) дроби; е) дата и время; ж) комплексные числа; з) разрежены матрицы переменной размерности, ненулевые элементы представлены односвязным списком. Разработать шаблонный класс (с использованием стандартной библиотеки шаблонов STL), реализующей определенную структуру хранения данных (т.н. "контейнерный" класс). Данные в контейнере хранятся либо в виде указателей на объекты, либо в виде самих объектов. Структуры хранения данных: а) двоичное дерево; б) массив; в) односвязный линейный список; г) двусвязный циклический список. Разработать демонстрационную программу</p> <p>Создание и использование динамически присоединяемых библиотек. Создание текстового редактора или калькулятора. Цель: освоить стандартную библиотеку шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. "</p> <p>Если задание предусматривает выполнение сортировки, ее необходимо выполнить с помощью алгоритма sort из STL. " Обработка ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Читать все строки входного файла в контейнер <code>vector<AnsiString></code>.</p>		
--	--	---	--	--

		<p>Отсортировать строки в контейнере в алфавитном порядке, преобразовать к верхнему регистру и записать их в выходной файл. Вариант 2. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Считать позиции начал строк входного файла в контейнер deque<int>. Используя эти данные и реверсивный итератор записать в выходной файл строки. Цель: углубить знания стандартной библиотеки шаблонов (STL). Общие требования: " для обращения к элементам заданного контейнера обязательно использовать соответствующие итераторы. " Если задание предусматривает выполнение сортировки, поиска или перебора диапазона, эти действия необходимо выполнить с помощью алгоритмов библиотеки или встроенных в контейнеры функций STL " Обработка ошибок должна быть реализована с помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. Страница - вектор из 24 строк, объектов класса AnsiString (последняя страница может быть неполной). Вывести на экран страницу с заданным номером. Вариант 2. typedef std::vector<AnsiString> Pages; std::list< Pages > Text; Считать все строки заданного входного файла в структуру данных - список страниц. С 1. Работа с инспектором объектов. " Создать приложение с главным окном, появляющимся в центре экрана, с размерами 200 на 300 (высота, ширина). " Сменить название класса формы на MForm, сменить заголовок формы. " Запретить изменение размеров окна и распахивание во весь экран. " Экспериментальным путем выяснить за что отвечает каждое из свойств класса TForm. " Нарисовать</p>		
--	--	---	--	--

		<p>собственную иконку используя редактор ресурсов и назначить ее приложению. 2. Обработка событий. "</p> <p>Доработать предыдущее приложение так, чтобы размеры и положение главной формы вычислялось автоматически при ее создании. Положение - правый нижний угол, высота и ширина равны четверти размеров экрана. (Событие OnCreate, объект Screen) "</p> <p>Поместить на форму стандартную кнопку. Сделать так, чтобы она "убегала" от курсора мыши. (Событие OnMouseMove, свойства Left, Top) 3. Диалоговая панель "</p> <p>Доработать приложение, поместив вторую кнопку и создав вторую форму. Сделать так, чтобы при нажатии на кнопку появлялась форма в модально</p>		
ПК.16/ОУ	<p>33. уметь использовать стандартные пакеты прикладных программ для решения практических задач</p>	<p>Создание текстового редактора или калькулятора.</p> <p>1. Работа с инспектором объектов. " Создать приложение с главным окном, появляющимся в центре экрана, с размерами 200 на 300 (высота, ширина). "</p> <p>Сменить название класса формы на TForm, сменить заголовок формы. " Запретить изменение размеров окна и распаивание во весь экран. "</p> <p>Экспериментальным путем выяснить за что отвечает каждое из свойств класс TForm. " Нарисовать собственную иконку используя редактор ресурсов и назначить ее приложению. 2. Обработка событий. "</p> <p>Доработать предыдущее приложение так, чтобы размеры и положение главной формы вычислялось автоматически при ее создании. Положение - правый нижний угол, высота и ширина равны четверти размеров экрана. (Событие OnCreate, объект Screen) "</p> <p>Поместить на форму стандартную кнопку. Сделать так, чтобы она "убегала" от курсора мыши. (Событие OnMouseMove, свойства Left, Top) 3. Диалоговая панель "</p> <p>Доработать приложение, поместив вторую кнопку и создав вторую форму. Сделать так, чтобы при нажатии на кнопку появлялась форма в</p>	<p>Курсовая работа</p> <p>Отчет по лабораторной работе, разделы...</p>	

		модально		
--	--	----------	--	--

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.9, ПК.16/ОУ.

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

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

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

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.9, ПК.16/ОУ, 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.

Форма билета на экзамен

Дисциплина *Программирование*

БИЛЕТ №

- 1 Вопрос...(1 – 23)
- 2 Вопрос...(24 – 47)

Составитель _____ В.В. Воронов

Заведующий кафедрой

_____ Е.В. Прохоренко
«___» _____ 20__ г.

СПИСОК ВОПРОСОВ

- 1. Понятие файла.
- 2. Файловая система.
- 3. Дескриптор и атрибуты файла.
- 4. Функции открытия и закрытия файла.
- 5. Функции записи и считывания.
- 6. Класс. Синтаксис, определение, атрибуты.
- 7. Элементы-данные и элементы-функции класса.
- 8. Объекты и их свойства. Время жизни.
- 9. Общие, частные и защищенные элементы класса.
- 10. Конструктор и деструктор. Свойства.
- 11. Конструкторы: копирования, с параметрами, без параметров. Область применения.
- 12. Конструктор копирования и оператор присваивания. Сходство и различия.
- 13. Типы объектов: внешний, автоматический, динамический, временный. Назначение и время жизни.
- 14. Переопределение операций.
- 15. Переопределение функций.
- 16. Типы результата: пустой, базовый, указатель и ссылка на объект, объект.
- 17. Друзья класса.
- 18. Иерархия классов. Базовый и производный классы.
- 19. Доступность элементов при наследовании.
- 20. Множественное наследование.
- 21. Виртуальные базовые классы.

22. Абстрактные классы.
23. Шаблоны функций.
24. Шаблоны классов.
25. Виртуальные функции. Чисто виртуальные функции.
26. Абстрактные классы и виртуальные деструкторы.
27. Понятия и исключения.
28. Перехватывание и управление исключениями.
29. Назначение и состав библиотеки STL
30. Контейнеры и итераторы.
31. Векторы.
32. Списки.
33. Ассоциативные списки. Понятие ключа и объекта.
34. Стеки и очереди.
35. Алгоритмы.
36. Структура приложения WINDOWS.
37. Краткая классификация классов в MFC.
38. Классы окон. Разновидности. Дескриптор окна.
39. Дочернее окно.
40. Всплывающее окно.
41. Цикл обработки сообщений.
42. Понятие ресурсов. Типы и создание ресурсов с помощью редактора.
43. Понятие DLL. Создание и присоединение DLL к программе.
44. Элементы окон: полосы прокрутки, панель состояния, инструментальная панель, иконки.
45. Понятие GDI Контексты устройств.
46. Графические классы и объекты. Построение геометрических фигур.
47. Разработка приложений с помощью App Wizard и Class Wizard.

Паспорт экзамена

по дисциплине «Программирование», 3 семестр

1. Методика оценки

Экзамен проводится в устной (письменной) форме, по билетам (тестам). Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-23, второй вопрос из диапазона вопросов 24-47(список вопросов приведен ниже). 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 Вопрос...(1 – 23)
2 Вопрос...(24 – 47)

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

2. Критерии оценки

- Ответ на билет для зачета считается неудовлетворительным, если студент при ответе не дает четкий ответ ни на один вопрос. Оценка составляет 0-49 баллов.
- Ответ на билет для зачета засчитывается на пороговом уровне, если студент дает ответ на оба вопроса, но есть некоторые неточности. Оценка составляет 50-72 баллов.
- Ответ на билет для зачета билет засчитывается на базовом уровне, если студент

дает ответ на один вопрос полностью, во втором есть некоторые недочеты. Оценка составляет 50-85 баллов.

- Ответ на билет (тест) для зачета билет засчитывается на продвинутом уровне, если студент дает полный ответ на оба вопроса. Оценка составляет 90-100 баллов.

3. Шкала оценки

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

4. Вопросы к экзамену по дисциплине «Программирование»

1. Понятие файла.
2. Файловая система.
3. Дескриптор и атрибуты файла.
4. Функции открытия и закрытия файла.
5. Функции записи и считывания.
6. Класс. Синтаксис, определение, атрибуты.
7. Элементы-данные и элементы-функции класса.
8. Объекты и их свойства. Время жизни.
9. Общие, частные и защищенные элементы класса.
10. Конструктор и деструктор. Свойства.
11. Конструкторы: копирования, с параметрами, без параметров. Область применения.
12. Конструктор копирования и оператор присваивания. Сходство и различия.
13. Типы объектов: внешний, автоматический, динамический, временный. Назначение и время жизни.
14. Переопределение операций.
15. Переопределение функций.
16. Типы результата: пустой, базовый, указатель и ссылка на объект, объект.
17. Друзья класса.
18. Иерархия классов. Базовый и производный классы.
19. Доступность элементов при наследовании.
20. Множественное наследование.
21. Виртуальные базовые классы.
22. Абстрактные классы.
23. Шаблоны функций.
24. Шаблоны классов.
25. Виртуальные функции. Чисто виртуальные функции.
26. Абстрактные классы и виртуальные деструкторы.
27. Понятия и исключения.
28. Перехватывание и управление исключениями.
29. Назначение и состав библиотеки STL
30. Контейнеры и итераторы.
31. Векторы.
32. Списки.
33. Ассоциативные списки. Понятие ключа и объекта.
34. Стеки и очереди.
35. Алгоритмы.
36. Структура приложения WINDOWS.
37. Краткая классификация классов в MFC.
38. Классы окон. Разновидности. Дескриптор окна.
39. Дочернее окно.

40. Всплывающее окно.
41. Цикл обработки сообщений.
42. Понятие ресурсов. Типы и создание ресурсов с помощью редактора.
43. Понятие DLL. Создание и присоединение DLL к программе.
44. Элементы окон: полосы прокрутки, панель состояния, инструментальная панель, иконки.
45. Понятие GDI Контексты устройств.
46. Графические классы и объекты. Построение геометрических фигур.
47. Разработка приложений с помощью App Wizard и Class Wizard

Критерии оценки

- Задание считается выполненным на **пороговом** уровне, если студент дает определение основных понятий, может показать причинно-следственные связи явлений, оценка составляет *50 - 73 балла*.
- Задание считается выполненным на **базовом** уровне, если студент формулирует основные понятия, законы, дает характеристику процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, оценка составляет *74 – 86 баллов*.
- Задание считается выполненным на **продвинутом** уровне, если студент проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, оценка составляет *87 - 100 баллов*.

Паспорт курсовой работы

по дисциплине «Программирование», 3 семестр

1. Методика оценки.

В рамках курсовой работы студент должен разработать многооконное приложение, используя библиотеку MFC

Содержание индивидуального задания:

- Задание;
- Разработка и отладка программного кода с максимальным использованием инструментов APP Wizard, Class Wizard и редактора ресурсов;
- Снабжение текста программы комментариями, а также выделение в тексте тех объектов и функций, которые введены программистом самостоятельно;
- Оформление отчета.

2. Критерии оценки.

- Задание считается выполненным на пороговом уровне, если студент дает определение основных понятий, может показать причинно-следственные связи явлений, оценка составляет 50 - 73 балла.
- Задание считается выполненным на базовом уровне, если студент формулирует основные понятия, законы, дает характеристику процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, оценка составляет 74 – 86 баллов.
- Задание считается выполненным на продвинутом уровне, если студент проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, 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.

4. Примерный перечень тем курсового проекта (работы).

1. Разработка приложения, которое отображает график заданной функции в заданном окне. Однодокументное приложение должно иметь системное меню, из которого вызывается диалоговая панель. В аналоговой панели должны выбираться тип графика (например, синусоида, меандр и экспонента) и их параметры (уровень, период и т.д.). В главном окне приложения создаются три дочерних окна. В системном меню должно быть подменю, в котором выбирается окно, где будет отражаться график заданной функции.
2. Приложение – небесный атлас созвездий. Приложение создает диалоговую панель, в которой имеется список участков звездного неба (северное и южное полушария,

зодиакальная область). При выборе участка неба программа автоматически выводит названия соответствующих созвездий.

В главном окне приложения создаются три дочерних окна. В системном меню должно быть подменю, из которого выбирается окно для вывода списка созвездий избранного участка неба. В приложении должно быть предусмотрено средство для добавления и удаления созвездия из выбранного списка и их редактирования.