

«

»

“ ”

“ ”
_____ .

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ Программирование

: 09.03.02

, :

: 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		

(): 09.03.02

219 12.03.2015 ., : 30.03.2015 .

: 1,

(): 09.03.02

, 2/1 20.06.2017

, 6 21.06.2017

:

, . .

:

,

:

. . .

1.

1.1

Компетенция ФГОС: ПК.28 монтажно-наладочная деятельность: способность к установке, отладке программных и настройке технических средств для ввода информационных систем в опытную и промышленную эксплуатацию; в части следующих результатов обучения:	
1.	
1.	
Компетенция ФГОС: ПК.3 способность проводить рабочее проектирование; в части следующих результатов обучения:	
2.	

2.

2.1

(
---	--

.3. 2	
1. О принципах объектно-ориентированного подхода при конструировании программ на языках высокого уровня.	;
2. О принципах программирования "под WINDOWS" с использованием библиотеки MFC.	;
.28. 1	
3. Основы языков высокого уровня C и C++.	;
4. Синтаксис и способы программирования задач на указанных языках.	;
5. Систему классов библиотеки MFC.	;
6. Принципы построения программ на базе MFC.	;
.28. 1	
7. Конструировать программы с помощью языка C++ и библиотеки MFC, направленные на инженерные приложения.	;
.28. 1	
8. Эффективно пользоваться электронным справочником MSDN.	;
.28. 1	
9. Читать, объяснять и корректировать тексты программ, направленных в среде C++.	;
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. Windows. Windows. Windows. Windows Windows Windows	0	2 5, 6
:			
17.		0	2 4, 5, 6
: MFC			
18.		0	1 4, 6
: DLL			
19.	DLL. DLL	0	2 4, 5
:			
20.		0	2 7
:			
21.		0	2 2, 6
: GDI-			
22.	(GDI).	0	2 5, 6
:			

5. (STL), " "). , . :); . 	2	2	7, 8, 9	- App Wizard, Class Wizard ;
6. : (STL). : " ." ," sort STL. " - Exception. " . 1. () (vector<AnsiString>. , . 2. () (deque<int>.	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>(, , ,</p> <p>) (, .).</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	0	0
<p>: . . - []:</p> <p>- / . . ; . . - . - , [2011]. -</p> <p>: http://courses.edu.nstu.ru/index.php?show=155&curs=351. -</p>				
4		2, 3	20	6
<p>: . . - []:</p> <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.

6.1

	.	
: 3		
Лабораторная:	20	40
Курсовая работа:	0	20
Экзамен:	20	40

6.2

6.2

		/	/	
.28	1.	+	+	+
	1.	+		+
.3	2.		+	+

1

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 "Znanium.com" : <http://znanium.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	(- , ,)	

1	(Internet)	

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)
ПК.28 монтажно-наладочная деятельность: способность к установке, отладке программных и настройке технических средств для ввода информационных систем в опытную и промышленную эксплуатацию	31. Знать особенности установки и отладки программных средств	<p>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 производные классы. Доступность их элементов. Наследование. Синтаксис конструирования объектов производных классов. Классы окон. Дочерние окна. Элементы окон: полосы прокрутки, панель состояния, инструментальная панель. 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</p>	Курсовая работа Отчет по лабораторной работе	Экзамен, вопросы 1-6

		<p>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 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. Поточный ввод-вывод в C++. 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. Создание и использование динамически присоединяемых</p>		
--	--	---	--	--

		<p>библиотек. Создание текстового редактора или калькулятора. Создание шаблона приложения. Создание однооконного приложения с помощью App Wizard. Создание многооконного приложения. Стандартный класс <code>std::string</code>. 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. Использование блоков <code>try/catch</code>. Неожиданные исключения и обработка завершения. 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 <code>sort</code> из 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. Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Читать все строки входного</p>		
--	--	---	--	--

		<p>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). 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;</code> <code>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;</code> <code>std::list< Pages > Text;</code> Читать все строки заданного входного файла в структуру данных - список страниц. С Шаблоны функций. Перегрузка и специализации шаблонов функций. Шаблоны функций сортировки. Шаблоны классов. Специализация шаблонов классов. Статические члены шаблонов классов. Класс - шаблон для вектора. 1. Работа с инспектором объектов. " Создать приложение с главным окном, появляющимся в центре</p>		
--	--	--	--	--

		<p>экрана, с размерами 200 на 300 (высота, ширина). "</p> <p>Сменить название класса формы на MForm, 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 модально</p>		
ПК.28	<p>у1. Уметь устанавливать и отлаживать различные программные средства</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 записи из двух полей</p>	Отчет по лабораторной работе	Экзамен, вопросы 7-12

		<p>[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, арифметических и сравнения)</p>		
--	--	---	--	--

		<p>- 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 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.</p> <p>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 помощью выбрасывания исключений - объектов класса Exception. " Работу программы проверить на файлах очень большого размера. Вариант 1.</p>		
--	--	---	--	--

		<p>Заданы имена входного текстового файла (существует на диске) и выходного файла (создается программой). Считать 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). Считать позиции начал строк входного файла в контейнер <code>deque<int></code>. Используя эти данные и реверсивный итератор записать в выходной файл стро. 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;</code> <code>std::list< Pages > Text;</code> Считать 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;</code> <code>std::list< Pages > Text;</code> 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 <code>MForm</code>, сменить</p>		
--	--	---	--	--

		<p>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 модально</p>		
ПК.3 способность проводить рабочее проектирование	32. знать инструменты и методы верификации структуры программного кода	<p>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.</p>	Курсовая работа,	Экзамен, вопросы 13-15

		Статические члены шаблонов классов. Класс - шаблон для вектора.		
--	--	---	--	--

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 ПК.28, ПК.3.

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 ПК.28, ПК.3, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

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

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

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

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

Паспорт экзамена

по дисциплине «Программирование», 3 семестр

1. Методика оценки

Экзамен проводится в устной (письменной) форме, по билетам (тестам). Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-7, второй вопрос из диапазона вопросов 8-15 (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. Фаза разработки, этапы процесса разработки. Стратегии конструирования ПО: линейная, инкрементная, эволюционная.
2. Диаграммы взаимодействия как элементы концептуальной модели. Синтаксис диаграмм взаимодействия.

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

(дата)

2. Критерии оценки

- Ответ на экзаменационный билет считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, оценка составляет *0-49 баллов*.
- Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, оценка составляет *50-72 баллов*.
- Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, законы, дает характеристику процессов, явлений, проводит анализ причин, условий, может представить

качественные характеристики процессов, оценка составляет 73-88 баллов.

- Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы 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 89-100 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. Фаза разработки, этапы процесса разработки. Стратегии конструирования ПО: линейная, инкрементная, эволюционная.
2. Стандарт ISO/IEC 12207-95: основные определения – система, модель жизненного цикла, квалификационные требования. Основные процессы, их содержание, работы и задачи процесса разработки.
3. Стандарт ISO/IEC 15504 (SPICE): оценка возможностей разработчика. Связь этого стандарта с моделью зрелости предприятия SEI CMM.
4. Прогностические модели процесса разработки: каскадная, RAD, спиральная.
5. Адаптивные модели процесса разработки: экстремальное программирование, Scrum.
6. Руководство программным проектом. Предварительные оценки проекта. Системный анализ и анализ требований. Анализ рисков. Планирование процесса разработки. Типовая структура распределения работ.
7. Контроль процесса разработки. Размерно- и функционально-ориентированные метрики. Метрические характеристики объектно-ориентированных систем.
8. Структурный и объектно-ориентированный подходы к разработке ПО. Их сравнительный анализ. Сущность объектного подхода к разработке программных средств.
9. Анализ предметной области: цели и задачи. Модели предметной области. Формальные определения. Классификация моделей. Методология IDEF0, синтаксис IDEF0-моделей.
10. Диаграммы потоков данных (DFD-диаграммы) и диаграммы потоков работ (IDEF3-диаграммы), их использование при моделировании предметной области.
11. Объектно-ориентированный анализ предметной области. Методика определения границ системы и ключевых абстракций. Пример проведения анализа. Функциональные и нефункциональные требования к системе.
12. Функциональные требования к системе. Способ их представления в виде UML-диаграммы. Пример диаграммы с использованием отношений «расширяет» и «включает». Понятие прецедента и сценария.
13. Концептуальная модель системы: концептуальные классы, системные события и системные операции. Способ их представления в виде UML-диаграмм. Пример концептуального описания прецедента.
14. Диаграммы взаимодействия как элементы концептуальной модели. Синтаксис диаграмм взаимодействия.
15. Проектирование программных средств. Цели и задачи этапа проектирования. Понятие модели проектирования, ее отличия от концептуальной модели. Стадии проектирования, их краткая характеристика.

Паспорт курсовой работы

по дисциплине «Программирование», 3 семестр

1. Методика оценки.

В работе студент должен написать программу в соответствии с индивидуальным заданием.

2. Критерии оценки.

- Работа считается **не выполненной**, если выполнены не все части РГЗ(Р), отсутствует анализ объекта, диагностические признаки не обоснованы, оценка составляет 0-49 баллов.
- Работа считается выполненной **на пороговом** уровне, если части РГЗ(Р) выполнены формально: анализ объекта выполнен без декомпозиции, диагностические признаки недостаточно обоснованы, , оценка составляет 50-72 баллов.
- Работа считается выполненной **на базовом** уровне, если анализ объекта выполнен в полном объеме, признаки и параметры диагностирования обоснованы, алгоритмы разработаны ,но не оптимизированы, оценка составляет 73-88 баллов.
- Работа считается выполненной **на продвинутом** уровне, если анализ объекта выполнен в полном объеме, признаки и параметры диагностирования обоснованы, алгоритмы разработаны и оптимизированы, оценка составляет 89-100 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 электронного словаря с возможностью подключения к базе данных.
2. создание программы для построения графиков математических функций на языке Object Pascal.
3. Создание программы для общения по локальной сети с выделенным сервером.
4. Разработка простейшего почтового клиента.
5. Разработка калькулятора логических функций.
6. Разработка компьютерной игры "Шарики"
7. Создание интерактивного приложения для выстраивания пользователем последовательностей из однотипных объектов в виде кристаллов.
8. Разработка интерактивной логической игры "Квест"
9. Создание модели летающего объекта "Вертикальная леталка"
10. Создание интерактивной модели поведения автомобиля "Гонки"
11. Создание интерактивной развивающей игры для детей "ПАЗЛ"

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

