


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Макаренко Елена Николаевна
Должность: Ректор
Дата подписания: 14.04.2018 22:21:39
Уникальный программный ключ:
c098bc0c1041cb2a4c6926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Ростовский государственный экономический университет (РИНХ)»



УТВЕРЖДАЮ
Первый проректор –
проректор по учебной работе

Н.Г. Кузнецов
«01» июня 2018г.

Рабочая программа дисциплины
Инженерное программирование

по профессионально-образовательной программе направление 09.03.04
"Программная инженерия"

Квалификация

Бакалавр

Ростов-на-Дону
2018 г.

КАФЕДРА Информационные технологии и защита информации

Распределение часов дисциплины по курсам

Курс	1		2		Итого	
	уп	рпд	уп	рпд		
Лекции	4	4	4	4	8	8
Лабораторные	6	6	6	6	12	12
В том числе инт.	6	6	6	6	12	12
Итого ауд.	10	10	10	10	20	20
Контактная	10	10	10	10	20	20
Сам. работа	94	94	125	125	219	219
Часы на контроль	4	4	9	9	13	13
Итого	108	108	144	144	252	252

ОСНОВАНИЕ

Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 09.03.04 "Программная инженерия" (уровень бакалавриата) (приказ Минобрнауки России от 12.03.2015г. №229)

Рабочая программа составлена

по профессионально-образовательной программе направление 09.03.04 "Программная инженерия"

Учебный план утвержден учёным советом вуза от 27.03.2018 протокол № 10.

Программу составил(и): к.э.н., доцент, Жилина Е.В.  11.05.18

Зав. кафедрой д.э.н., проф. Тищенко Е.Н.  11.05.18

Методическим советом направления к.ф.-м.н., декан, Карасев Д.Н.  16.05.18

Отделом образовательных программ и планирования учебного процесса Торопова Т.В.  30.05.18

Проректором по учебно-методической работе Джуха В.М.  31.05.18

**Визирование РПД для исполнения в очередном учебном
году**

Отдел образовательных программ и планирования
учебного процесса Торопова Т.В. _____

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2019-2020 учебном году на заседании
кафедры **Информационные технологии и защита информации**

Зав. кафедрой д.э.н., проф. Тищенко Е.Н. _____

Программу составил(и) *к.э.н., доцент, Жилина Е.В.* _____

**Визирование РПД для исполнения в очередном учебном
году**

Отдел образовательных программ и планирования
учебного процесса Торопова Т.В.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2020-2021 учебном году на заседании
кафедры **Информационные технологии и защита информации**

Зав. кафедрой д.э.н., проф. Тищенко Е.Н. _____

Программу составил(и): *к.э.н., доцент, Жилина Е.В.* _____

**Визирование РПД для исполнения в очередном учебном
году**

Отдел образовательных программ и планирования
учебного процесса Торопова Т.В.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2021-2022 учебном году на заседании
кафедры **Информационные технологии и защита информации**

Зав. кафедрой д.э.н., проф. Тищенко Е.Н. _____

Программу составил(и): *к.э.н., доцент, Жилина Е.В.* _____

**Визирование РПД для исполнения в очередном учебном
году**

Отдел образовательных программ и планирования
учебного процесса Торопова Т.В.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2022-2023 учебном году на заседании
кафедры **Информационные технологии и защита информации**

Зав. кафедрой д.э.н., проф. Тищенко Е.Н. _____

Программу составил(и): *к.э.н., доцент, Жилина Е.В.* _____

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	Цель освоения дисциплины: является изучение парадигм и методов инженерного программирования, ознакомление с методами разработки программ и основными приемами программирования на языке C++ с помощью инструментов разработки программ (MS Visual Studio) для овладения знаниями в области программирования инженерных подсистем (модулей).
1.2	Задачи освоения дисциплины: получение навыков владения инструментами и методами обработки структурных и объектных данных; пролучение практических навыков разработки алгоритмов, подбора адекватных задаче структур инженерных данных и их реализации на современных программных средствах.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ООП:	Б1.Б
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Для успешного освоения дисциплины студент должен иметь базовую подготовку по Информатике в объеме средней школы
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Практика по получению профессиональных умений и опыта профессиональной деятельности
2.2.2	Методы отказоустойчивого программирования
2.2.3	Методы разработки защищенных систем
2.2.4	Практика по получению профессиональных умений и опыта профессиональной деятельности
2.2.5	Обеспечение качества и тестирование программного обеспечения
2.2.6	Реинжиниринг систем программирования
2.2.7	Итоговая государственная аттестация
2.2.8	Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности
2.2.9	Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности
2.2.10	Практика по получению профессиональных умений и опыта профессиональной деятельности
2.2.11	Подготовка к сдаче и сдача государственного экзамена
2.2.12	Реинжиниринг систем программирования

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
ОПК-1: владением основными концепциями, принципами, теориями и фактами, связанными с информатикой	
Знать:	
способы, методы решения стандартных задачи профессиональной деятельности	
Уметь:	
решать стандартные задачи профессиональной деятельности с применением информационно-коммуникационных технологий	
Владеть:	
навыками решения стандартных задач профессиональной деятельности с применением информационно-коммуникационных технологий и с учетом основных требований программной инженерии	
ОПК-3: готовностью применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	
Знать:	
основы информатики и программирования	
Уметь:	
применять методы составления алгоритмов решения инженерных задач	
Владеть:	
навыками работы в CASE-средствах проектирования и разработки программных продуктов	
ПК-3: владением навыками использования различных технологий разработки программного обеспечения	
Знать:	
основы применения технологий разработки программного обеспечения	
Уметь:	
использовать различные технологии разработки программного обеспечения	
Владеть:	

навыками разработки инженерного программного обеспечения
ПК-21: владением навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации
Знать:
базовые алгоритмы решения инженерных задач
Уметь:
выделять главную идею прочитанного исходного кода
Владеть:
навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Интер акт.	Примечание
Раздел 1. 1. «Структуры данных»							
1.1	Тема 1. «Типы данных в языке C++» Понятие типов данных в языках программирования, классификация типов дан-ных в C++, особенности представления базовых типов и операций над ними, рекомендации и правила выполнения операции преобразования базовых типов в C++. Подставляемые (встраиваемые) функции. Перегрузка функций. Рекурсивные функции /Лек/	1	2	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	1	
1.2	Тема 1. «Типы данных в языке C++» Типы данных в языке C++ Подставляемые (встраиваемые) функции. Перегрузка функций. Рекурсивные функции. /Лаб/	1	2	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	2	
1.3	Тема 2. «Производные типы. Указатели на объекты, указатели на функции» Производные типы. Тип указатель: указатели на объекты. Тип указатель: указатели на функции. Решение задач с использованием указателей. /Ср/	1	8	ОПК-1 ОПК-3 ПК-3	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
1.4	Тема 1. «Типы данных в языке C++» Символьные данные и строки: понятия и определения символьных данных и строк, сходство и отличия их внутреннего представления, способы объявления, инициализация строк, методы доступа к элементам строк, определение размера строк, различные способы организации ввода/вывода символьных данных и строк. /Ср/	1	8	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
1.5	Тема 2. «Производные типы. Указатели на объекты, указатели на функции» Определение и виды указателей, способы объявления, инициализация указателей на объекты, методы доступа к данным через указатели, размещение указателей и адресуемых ими объектов в памяти, операции над указателями. Тип указатель: указатели на функции. Решение задач с использованием указателей /Ср/	1	8	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	

1.6	Тема 2. «Производные типы. Указатели на объекты, указатели на функции» Функции для работы со строками: библиотечные функции, предназначенные для работы со строками, и особенности их выполнения, приведены примеры использования таких функций при решении задач. /Ср/	1	6	ОПК-1 ОПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
1.7	Тема 3. «Массивы: одномерные и двумерные» Массивы: одномерные массивы, задачи поиска, замены и перестановок элементов массива, задачи сортировок элементов массива. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива, задачи сортировок и перестановок в двумерных массивах. /Лаб/	1	2	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	1	
1.8	Тема 3. «Массивы: одномерные и двумерные» Понятие и виды массивов в языках программирования, определение, объявление, инициализация, способы генерации и вывод одномерных массивов, расположение в памяти элементов массивов, связь между указателями и массивами, задачи поиска, замены и перестановок элементов массива, сортировок элементов массива. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива, задачи сортировок и перестановок в двумерных массивах /Лек/	1	2	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	1	
1.9	Тема 3. «Массивы: одномерные и двумерные» Битовые поля: объявление и размещение в памяти битовых полей, методы доступа к битовым полям, приемы использования битовых полей для побитового анализа данных. /Ср/	1	4	ОПК-1 ОПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
1.10	Тема 4. «Структуры» Определения, способы объявления, инициализация структур, методы доступа к данным структуры, размещение структур и определение их размера в памяти, массивы структур. Структуры и указатели. Объединения. /Ср/	1	4	ОПК-1 ОПК-3 ПК-3	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
1.11	Тема 4. «Структуры» Алгоритмы сжатия данных: основные понятия и алгоритмы сжатия данных, приводятся примеры программной реализации алгоритма Хаффмана через префиксные коды и на основе кодовых деревьев. /Ср/	1	6	ОПК-1 ОПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	

1.12	Тема 5. «Стандартные файлы и функции по работе с ними» Понятия и определения файлов и потоков, классификация файлов, стандартные функции по работе с файлами, основные алгоритмы работы с файлами, примеры решения задач на организацию ввода-вывода и обработку данных в файлах. Обработка файлов в потоковом режиме. Прямой доступ к данным файла. Работа с элементами файлов. Решение задач на обработку файлов. /Ср/	1	4	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
1.13	Тема 5. «Стандартные файлы и функции по работе с ними» Стандартные файлы и функции по работе с ними. Обработка файлов в потоковом режиме. Прямой доступ к данным файла. Работа с элементами файлов. Решение задач на обработку файлов. /Ср/	1	4	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
	Раздел 2. 2. «Алгоритмы обработки данных»						
2.1	Тема 6. «Распределение памяти. Динамическое выделение памяти» Решение задач на динамическое выделение памяти. Массивы. Обработка элементов массива. Диагонали. Передача массива в функцию. /Лаб/	1	2	ОПК-1 ОПК-3 ПК-3	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	1	
2.2	Тема 6. «Распределение памяти. Динамическое выделение памяти» Распределение памяти. Динамическое выделение памяти. Одномерные динамические массивы. Двумерные динамические массивы. Решение задач на динамические массивы. Динамические структуры данных: однонаправленные и двунаправленные списки, очередь и стек, бинарные деревья /Ср/	1	8	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
2.3	Тема 6. "Распределение памяти. Динамическое выделение памяти" Составление алгоритмов по обработке динамических строк /Ср/	1	4	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
2.4	Тема 7. «Алгоритмы обработки данных» Понятие ресурсной эффективности алгоритмов посредством анализа асимптотических функций временной и емкостной сложности, классификация алгоритмов на основе функции временной сложности, робщие методы оценки трудоемкости алгоритмов. Рекурсия и рекурсивные алгоритмы. Алгоритм перебора с возвратом. Алгоритмы поиска в линейных структурах. Алгоритмы хеширования данных /Ср/	1	6	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	

2.5	Тема 8. «Алгоритмы поиска на основе деревьев» Определение и виды деревьев поиска, приемы снижения трудоемкости поиска в древовидных структурах, описания алгоритмов поиска в двоичных упорядоченных, случайных и сбалансированных в высоту (АВЛ) деревьях, примеры программной реализации бинарного дерева поиска и АВЛ-дерева. /Ср/	1	4	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
2.6	Тема 8. «Алгоритмы поиска на основе деревьев» Тема 5. Алгоритмы на графах. Алгоритмы обхода графа: основные понятия из теории графов, модели представления графов, на основе которых приводятся описания и реализации алгоритмов поиска в глубину и в ширину. /Ср/	1	6	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
2.7	Тема 9. «Алгоритмы сортировки массивов» Определение и классификация алгоритмов сортировок массивов, в частности, быстрых сортировок, изучаются параметры, характеризующие трудоемкость алгоритмов сортировок, рассматриваются описания и примеры программных кодов следующих алгоритмов быстрых сортировок: бинарная пирамидальная сортировка, сортировка слиянием, сортировка Шелла и сортировка Хоара. Внутренняя сортировка. Внешняя сортировка /Ср/	1	6	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
2.8	Тема 9. «Алгоритмы сортировки массивов» Алгоритмы на графах. Алгоритмы нахождения кратчайшего пути: постановка задачи и описание алгоритмов нахождения кратчайшего пути в графах, программные реализации алгоритмов Дейкстры, Флойда и переборного алгоритма. /Ср/	1	8	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
2.9	/Зачёт/	1	4	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
	Раздел 3.3. «Языки программирования»						

3.1	Тема 10 «Развитие языков программирования» История ЯП. Классификация ЯП. Области применения ЯП. Парадигмы программирования. Процедурные языки. Аппликативные языки. Языки логического программирования. Объектно-ориентированные языки. Стандартизация языков программирования. Среда проектирования Visual Studio.NET. /Ср/	2	4	ОПК-1 ОПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
3.2	Тема 11. «Основные понятия объектно-ориентированного программирования (ООП)» Объектно-ориентированное программирование. Объект. Класс. Инкапсуляция. Наследование. Полиморфизм. Секции private, protected, public. Указатель this. Переопределение операций. /Лек/	2	2	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	1	
3.3	Тема 11. «Основные понятия объектно-ориентированного программирования (ООП)» Программирование разветвляющегося алгоритма. Разработка программного кода с циклическим алгоритмом. Разработка программы-класса, содержащей атрибуты, методы класса. Написать программу нахождение произведения. Свойства класса – две переменные. Метод класса – функция нахождения произведения. Написать программу нахождение суммы. Свойства класса – три переменные. Метод класса – функция нахождения суммы. /Лаб/	2	2	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	2	
3.4	Тема 11. «Основные понятия объектно-ориентированного программирования (ООП)» Преобразование типов. static_cast. dynamic_cast /Ср/	2	2	ОПК-1 ОПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
3.5	Тема 12. «Компоновка файлов в одну программу» Реализовать объединение нескольких проектов в одно решение (mix). Класс Sum использовать 3 раза в разных файлах исходного кода двух проектов. /Лаб/	2	2	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	1	
3.6	Тема 12. «Компоновка файлов в одну программу» Поток и is_open(). Передача параметров через консоль (int argc, char *argv[]). Режимы файла. Бинарные файлы. Решение. Проект. Директивы #include, #define, #undef, #ifdef, #else, #if, #pragma. Макроимена. __FILE__ и __LINE__. Условная компиляция. /Ср/	2	7	ОПК-1 ОПК-3 ПК-3	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	

3.7	Тема 12. «Компоновка файлов в одну программу» Сдвинуть элементы массива на m позиций влево /Ср/	2	8	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
Раздел 4. 4. «Практическое инженерное программирование»							
4.1	Тема 13. «Конструкторы и деструкторы. Инкапсуляция» Разработка программы с конструктором с использованием шаблона. Инициализация свойств. Const метод класса. Разработка программы с конструктором без использования шаблона. Передача параметров в программу. /Лаб/	2	2	ОПК-1 ОПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	1	
4.2	Тема 13. «Конструкторы и деструкторы. Инкапсуляция» Конструктор. Инициализация переменных в конструкторе. Конструктор без параметров. Копирующий конструктор. Содержательный конструктор. Константные ссылки и указатели. Константные методы классов. Перегрузка. Инкапсуляция. /Ср/	2	8	ОПК-1 ОПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
4.3	Тема 13. «Конструкторы и деструкторы. Инкапсуляция» Написать программу вычисления площади треугольника по формуле Герона ($S = \sqrt{p(p-a)(p-b)(p-c)}$, где p — полупериметр треугольника), используя методы класса Площадь и Периметр /Ср/	2	4	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
4.4	Тема 14. «Наследование» Понятие наследования. Виды наследования. Множественное наследование. Особенности. /Лек/	2	2	ОПК-1 ОПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	1	
4.5	Тема 14. «Наследование» Реализуйте класс Vectors, содержащий 2 конструктора, деструктор и функцию определения длины вектора в трёхмерном пространстве через координаты x, y, z ($d = \sqrt{x^2 + y^2 + z^2}$) /Ср/	2	8	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
4.6	Тема 15. «Дружественные функции. Дружественные классы» Дружественные функции. Дружественные члены класса (методы). Дружественные классы. Объявление класса. /Ср/	2	4	ОПК-1 ОПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
4.7	Тема 15. «Дружественные функции. Дружественные классы» Создайте класс, который будет выполнять вывод всех чисел в HEX формате и без пробелов /Ср/	2	4	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	

4.8	Тема 16. «Виртуальные методы. Абстрактные классы. Обработка исключений» Виртуальные методы. Виртуальная функция. Виртуальные классы. Абстрактные классы. Виды ошибок. Исключительные ситуации. Обработка исключительных ситуаций. /Ср/	2	10	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
4.9	Тема 16. «Виртуальные методы. Абстрактные классы. Обработка исключений» Программа должна содержать перегруженную операцию «=», использование которой копирует S1 в S2 при условии: без 2 первых и 2 последних символа /Ср/	2	4	ОПК-1 ОПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
4.10	Тема 17. «Стандартная библиотека STL» Шаблоны методов. Шаблоны классов. Назначение шаблонов. Функции-шаблоны. «Интеллектуальный указатель». Задание свойств класса. /Ср/	2	8	ОПК-1 ОПК-3 ПК-3	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	
4.11	Тема 17. «Виртуальные методы. Абстрактные классы. Обработка исключений» STL <algorithm> <vector>, <list> /Ср/	2	18	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.2 Л3.3 Л3.4 Э1 Э2	0	
4.12	Контрольная работа. Перечень тем представлен в Приложении 1 к рабочей программе дисциплины. /Ср/	2	36	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4	0	
4.13	/Экзамен/	2	9	ОПК-1 ОПК-3 ПК-3 ПК-21	Л1.1 Л1.2 Л1.3 Л1.4 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Л3.4 Э1 Э2 Э3 Э4	0	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Фонд оценочных средств для проведения промежуточной аттестации

Вопросы к зачету, 1 курс:

1. Понятие типов данных в языках программирования, классификация типов данных в C++
2. Особенности представления базовых типов и операций над ними
3. Рекомендации и правила выполнения операции преобразования базовых типов в C++.
4. Подставляемые (встраиваемые) функции. Перегрузка функций. Рекурсивные функции.
5. Определение и виды указателей, способы объявления, инициализация указателей на объекты.
6. Методы доступа к данным через указатели, размещение указателей и адресуемых ими объектов в памяти, операции над указателями.
7. Понятие и виды массивов, определение, объявление, инициализация, способы генерации и вывод одномерных массивов, расположение в памяти элементов массивов, связь между указателями и массивами
8. Задачи поиска, замены и перестановок элементов массива, сортировок элементов массива.
9. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива, задачи сортировок и перестановок в двумерных массивах.
10. Определения, способы объявления, инициализация структур, методы доступа к данным структуры, размещение структур и определение их размера в памяти, массивы структур.
11. Структуры и указатели.
12. Объединения.
13. Понятия и определения файлов и потоков, классификация файлов, стандартные функции по работе с файлами,

основные алгоритмы работы с файлами.

14. Обработка файлов в потоковом режиме. Прямой доступ к данным файла. Работа с элементами файлов.
15. Распределение, способы выделения и освобождения динамической памяти, взаимодействие указателей и участков динамической памяти.
16. Одномерные динамические массивы. Двумерные динамические массивы.
17. Динамические структуры данных: однонаправленные и двунаправленные списки, очередь и стек, бинарные деревья.
18. Понятие ресурсной эффективности алгоритмов посредством анализа асимптотических функций временной и емкостной сложности.
18. Классификация алгоритмов на основе функции временной сложности, общие методы оценки трудоемкости алгоритмов.
19. Рекурсия и рекурсивные алгоритмы.
20. Алгоритм перебора с возвратом.
21. Алгоритмы поиска в линейных структурах.
22. Алгоритмы хеширования данных.
23. Определение и виды деревьев поиска, приемы снижения трудоемкости поиска в древовидных структурах.
24. Алгоритмы поиска в двоичных упорядоченных, случайных и сбалансированных в высоту (АВЛ) деревьях. Примеры программной реализации бинарного дерева поиска и АВЛ-дерева.
25. Определение и классификация алгоритмов сортировок массивов.
26. Бинарная пирамидальная сортировка.
27. Сортировка слиянием.
28. Сортировка Шелла.
29. Сортировка Хоара.
30. Внутренняя сортировка. Внешняя сортировка.

Вопросы к экзамену, 2 курс:

1. История ЯП. Классификация и область применения ЯП.
2. Парадигмы программирования. Процедурные языки. Аппликативные языки.
3. Языки логического программирования. Объектно-ориентированные языки.
4. Стандартизация языков программирования.
5. Среда проектирования VisualStudio.NET. Типы данных.
6. Поточный ввод\вывода данных printf() и scanf(). Библиотека stdio.h. Операторы ввода\вывода (cout и cin). Библиотека <iostream>. Команда using namespace std.
7. Операция увеличения ++ и уменьшения --: «префиксная» форма, и «постфиксная» форма. Проверка четности числа. Логические операции. Очистка экрана. Файла conio.h.
8. Программирование разветвляющихся алгоритмов. Условные операторы. Оператор выбора switch .
9. Функции библиотеки math.h. Линейный алгоритм.
10. Программирование циклических алгоритмов.
11. Программирование с использованием одномерных массивов. Заполнение матрицы случайными числами.
12. Объектно-ориентированное программирование. Объект. Класс.
13. Инкапсуляция.
14. Полиморфизм.
15. Секции private, protected, public. Указатель this.
16. Переопределение операций.
17. Потоки работы с файлами. Поток и is_open().
18. Передача параметров через консоль (int argc, char *argv[]). Режимы файла. Бинарные файлы.
19. Решение. Проект.
20. Директивы #include, #define, #undef, #ifdef, #else, #if, #pragma.
21. Макроимена. __FILE__ и __LINE__.
22. Конструктор. Инициализация переменных в конструкторе. Конструктор без параметров. Копирующий конструктор. Содержательный конструктор.
23. Константные ссылки и указатели. Константные методы классов. Перегрузка.
24. Понятие наследования. Виды наследования.
25. Множественное наследование.
26. Дружественные функции. Дружественные члены класса (методы). Дружественные классы.
27. Виртуальные методы. Виртуальная функция. Виртуальные классы.
28. Абстрактные классы.
29. Виды ошибок. Исключительные ситуации. Обработка исключительных ситуаций. Операторы try и catch.
30. Шаблоны методов. Шаблоны классов. Назначение шаблонов.

5.2. Фонд оценочных средств для проведения текущего контроля

Структура и содержание фонда оценочных средств представлены в Приложении 1 к рабочей программе дисциплины.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

Авторы, составители	Заглавие	Издательство, год	Колич-во
---------------------	----------	-------------------	----------

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л1.1	Кетков Ю. Л.	Введение в языки программирования С и С++: курс	Москва: Интернет-Университет Информационных Технологий, 2008	http://biblioclub.ru/ - неограниченный доступ для зарегистрированных пользователей
Л1.2	Страуструп Б.	Язык программирования С++ для профессионалов	Москва: Интернет-Университет Информационных Технологий, 2006	http://biblioclub.ru/ - неограниченный доступ для зарегистрированных пользователей
Л1.3	Мейер Б.	Объектно-ориентированное программирование и программная инженерия	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	http://biblioclub.ru/ - неограниченный доступ для зарегистрированных пользователей
Л1.4	Сорокин А. А.	Объектно-ориентированное программирование: учебное пособие (курс лекций)	Ставрополь: СКФУ, 2014	http://biblioclub.ru/ - неограниченный доступ для зарегистрированных пользователей

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л2.1	Савельева Н. Г., Веретенникова Е. Г.	Информатика и программирование: учеб. пособие	Ростов н/Д: Изд-во РГЭУ (РИНХ), 2016	64
Л2.2	Зольников В. К., Машевич П. Р., Анциферова В. И., Литвинов Н. Н.	Программирование и основы алгоритмизации: учебное пособие	Воронеж: Воронежская государственная лесотехническая академия, 2011	http://biblioclub.ru/ - неограниченный доступ для зарегистрированных пользователей
Л2.3	Агафонов Е. Д., Ващенко Г. В.	Прикладное программирование: учебное пособие	Красноярск: Сибирский федеральный университет, 2015	http://biblioclub.ru/ - неограниченный доступ для зарегистрированных пользователей

6.1.3. Методические разработки

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л3.1	Тищенко Е. Н., Жилина Е. В.	Эффективная работа в MS Office: практикум	Ростов н/Д: Изд-во РГЭУ (РИНХ), 2015	63
Л3.2	Савельева Н. Г., Веретенникова Е. Г., Мирошниченко И. И., Гречкина В. Ю.	Работа в среде Visual Studio 2013 (С++): лаборатор. практикум по программированию	Ростов н/Д: Изд-во РГЭУ (РИНХ), 2014	53
Л3.3	Савельева Н. Г., Жилина Е. В.	Работа в среде Visual Studio 2013 (С++): лаборатор. практикум по объектно-ориентир. программированию	Ростов н/Д: Изд-во РГЭУ (РИНХ), 2014	63
Л3.4	Комлева Н. В., Ковалевская Е. В.	Методы программирования: учебно-методический комплекс	Москва: Евразийский открытый институт, 2011	http://biblioclub.ru/ - неограниченный доступ для зарегистрированных пользователей

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"


Э1	Гагарина, Л. Г. Алгоритмы и структуры данных [Электронный ресурс]: учебное пособие / Л. Г. Гагарина, В. Д. Колдаев. - М.: Финансы и статистика, 2009. - 304 с. http://biblioclub.ru/index.php?page=book&id=225965			
Э2	Информатика. Математическое и программное обеспечение. В 3-х ч. Ч. 1. Модели решения функциональных и вычислительных задач. Учебное пособие. [Электронный ресурс]. - Воронеж: Воронежская государственная лесотехническая академия, 2007, 128 с. http://biblioclub.ru/index.php?page=book&id=143299			
Э3	Информатика. Математическое и программное обеспечение. В 3-х ч. Ч. 2. Технологии программирования. Учебное пособие. [Электронный ресурс]. - Воронеж: Воронежская государственная лесотехническая академия, 2007, 208 с. http://biblioclub.ru/index.php?page=book&id=143300			
Э4	Гров Ю.Ю. , Дидрих В.Е. , Иванова О.Г. , Однолько В.Г. Теория информационных процессов и систем: учебное пособие. – [Электронный ресурс]. - Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2014. – 172 с. http://biblioclub.ru/index.php?page=book&id=277939&sr=1			

6.3. Перечень программного обеспечения	
6.3.1	MS Visual Studio
6.3.2	MS Visio
6.4 Перечень информационных справочных систем	
6.4.1	Консультант +
6.4.2	Гарант

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)	
7.1	Помещения для проведения всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения. Для проведения лекционных занятий используется демонстрационное оборудование. Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными программными средствами и выходом в Интернет.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)	
Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.	

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ростовский государственный экономический университет (РИНХ)»

Рассмотрено и одобрено
на заседании кафедры Информационные
технологии и защита информации
Протокол № 10 от 11.05.2018 г.
Зав.кафедрой  Тищенко Е.Н.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО ДИСЦИПЛИНЕ**

Инженерное программирование

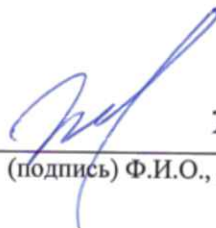
Направление подготовки

09.03.04 Программная инженерия

Уровень образования

Бакалавриат

Составитель



Жилина Е.В., доцент, к.э.н.

(подпись) Ф.И.О., должность, ученая степень, ученое звание

Ростов-на-Дону, 2018

Оглавление

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.....	3
2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....	3
3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	5
4. Методические материалы, определяющие процедуры оценивания результатов освоения образовательной программы	17

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Перечень компетенций с указанием этапов их формирования представлен в п. 3. «Требования к результатам освоения дисциплины» рабочей программы дисциплины.

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

2.1 Показатели и критерии оценивания компетенций:

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
ОПК 1 - владением основными концепциями, принципами, теориями и фактами, связанными с информатикой			
З. способы, методы решения стандартных задачи профессиональной деятельности	актуальность темы исследования и ее научно-практическая новизна	полнота и содержательность ответа умение приводить примеры	КО – контрольный опрос, РЗ – расчетные задачи, ЛЗ – лабораторные задания, КР – контрольная работа, ИЗ – индивидуальное задание
У. решать стандартные задачи профессиональной деятельности с применением информационно-коммуникационных технологий	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
В. навыками решения стандартных задач профессиональной деятельности с применением информационно-коммуникационных технологий и с учетом основных требований программной инженерии	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
ОПК 3 - готовностью применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов			
З. основы информатики и программирования	актуальность темы исследования и ее научно-практическая новизна	полнота и содержательность ответа умение приводить примеры	КО – контрольный опрос, РЗ – расчетные задачи, ЛЗ – лабораторные задания, КР – контрольная работа, ИЗ – индивидуальное задание
У. применять методы составления алгоритмов решения инженерных задач	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
В. навыками работы в CASE-средствах проектирования и разработки программных продуктов	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
ПК 3 - владением навыками использования различных технологий разработки программного обеспечения			
З. основы применения технологий разработки	актуальность темы исследования и ее научно-	полнота и содержательность ответа	КО – контрольный опрос, РЗ – расчетные задачи, ЛЗ

программного обеспечения	практическая новизна	умение приводить примеры	– лабораторные задания, КР – контрольная работа, ИЗ – индивидуальное задание
У. использовать различные технологии разработки программного обеспечения	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
В. навыками разработки инженерного программного обеспечения	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
ПК 21 - владением навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации			
З. базовые алгоритмы решения инженерных задач	актуальность темы исследования и ее научно-практическая новизна	полнота и содержательность ответа умение приводить примеры	КО – контрольный опрос, РЗ – расчетные задачи, ЛЗ – лабораторные задания, КР – контрольная работа, ИЗ – индивидуальное задание
У. выделять главную идею прочитанного исходного кода	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
В. навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	

2.2 Шкалы оценивания:

Текущий контроль успеваемости и промежуточная аттестация осуществляется в рамках накопительной балльно-рейтинговой системы в 100-балльной шкале

2 семестр – очная форма обучения, 1 курс - заочная форма обучения:

- 50-100 баллов («зачет») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой; наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины; наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов («незачет») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

3 семестр – очная форма обучения, 2 курс (заочная форма обучения):

- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка «удовлетворительно») – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка «неудовлетворительно») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

В разделе приводятся типовые варианты оценочных средств: вопросы к контрольному опросу, вопросы к зачету, экзамену, лабораторные задания, расчетные задачи, тематика индивидуальных заданий (контрольной работы).

Вопросы к контрольному опросу, 3 семестр (очная форма обучения) по дисциплине Инженерное программирование

Раздел 3 «Языки программирования»:

Контрольный письменный опрос по следующим вопросам:

Вариант 1

Теоретические задания:

1. Классификация ЯП.
2. Аппликативные языки.
3. Объектно-ориентированные языки.
4. Операторы ввода\вывода (cout и cin).
5. Функции библиотеки math.lib.
6. Оператор выбора switch.
7. Класс.
8. Указатель this.

Практические задания:

1. Напишите программу по заполнению двумерной матрицы случайными числами.
2. Напишите программу нахождения N:

$$H = \begin{cases} z+t, & \text{если } z < 5 \\ z/t, & \text{если } z+t > 50 \\ 2z-t, & \text{иначе} \end{cases}$$

3. Напишите класс Товар. Структуру класса продумайте самостоятельно.

Вариант 2

Теоретические задания:

1. Области применения ЯП.
2. Процедурные языки.
3. Языки логического программирования.
4. Функции потокового ввода\вывода данных printf() и scanf().
5. Функции библиотеки math.lib.
6. Программирование циклических алгоритмов.
7. Объект.
8. Секции private, protected, public.

9. Практические задания:

1. Напишите программу по заполнению одномерной матрицы случайными числами.
2. Напишите программу нахождения F:

$$F = \begin{cases} a+b, & \text{если } a < b \\ a/b, & \text{если } a > b \\ a \times b, & \text{иначе} \end{cases}$$

3. Напишите класс Студент. Структуру класса продумайте самостоятельно.

Раздел 4 «Практическое программирование»:

Контрольный письменный опрос по следующим вопросам:

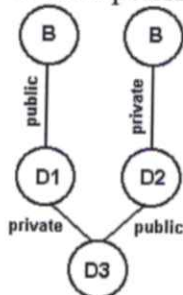
Вариант 1

Теоретические задания:

1. Опишите понятие конструктора.
2. Инкапсуляция. Принципы построения программного кода.
3. Дружественные функции. Особенности, использование.
4. Построение виртуальных классов.

Практические задания:

1. Построить иерархию классов согласно схеме наследования



2. Разработайте пользовательский класс **Arr**, который содержит:

- Конструктор **Arr ()** с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов (размеры массива - число строк и столбцов передается в конструктор через параметры);
- Деструктор: **~ Arr ()** освобождения памяти;
- Метод печати текущего состояния массива: **void Print (...)**;
- Функция-друг: **friend void Run (...)**, решающая: количество строк, содержащих хотя бы один нулевой элемент.

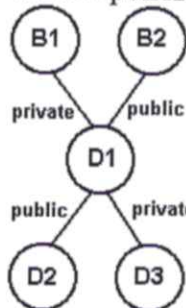
Вариант 2

Теоретические задания:

1. Опишите понятие деструктора.
2. Полиморфизм. Принципы построения программного кода.
3. Дружественные классы. Особенности, использование.
4. Построение абстрактных классов.

Практические задания:

1. Построить иерархию классов согласно схеме наследования



2. Разработайте пользовательский класс **Mas**, который содержит:

- Конструктор **Mas ()** с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов (размеры массива - число строк и столбцов передается в конструктор через параметры);
- Метод печати текущего состояния массива: **void Vyvod (...)**;
- Метод переустановки текущего состояния массива (размера массива): **void Set (...)**;
- Функция-друг **friend void Run (...)**, решающая индивидуальную задачу: номер первой из строк, не содержащих ни одного положительного элемента.

Вопросы к зачету, 2 семестр (очная форма обучения), 1 курс (заочная форма обучения) по дисциплине Инженерное программирование

1. Понятие типов данных в языках программирования, классификация типов данных в C++
2. Особенности представления базовых типов и операций над ними
3. Рекомендации и правила выполнения операции преобразования базовых типов в C++.
4. Подставляемые (встраиваемые) функции. Перегрузка функций. Рекурсивные функции.
5. Определение и виды указателей, способы объявления, инициализация указателей на объекты.
6. Методы доступа к данным через указатели, размещение указателей и адресуемых ими объектов в памяти, операции над указателями.
7. Понятие и виды массивов, определение, объявление, инициализация, способы генерации и вывод одномерных массивов, расположение в памяти элементов массивов, связь между указателями и массивами
8. Задачи поиска, замены и перестановок элементов массива, сортировок элементов массива.
9. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива, задачи сортировок и перестановок в двумерных массивах.
10. Определения, способы объявления, инициализация структур, методы доступа к данным структуры, размещение структур и определение их размера в памяти, массивы структур.
11. Структуры и указатели.
12. Объединения.

13. Понятия и определения файлов и потоков, классификация файлов, стандартные функции по работе с файлами, основные алгоритмы работы с файлами.

14. Обработка файлов в потоковом режиме. Прямой доступ к данным файла. Работа с элементами файлов.

15. Распределение, способы выделения и освобождения динамической памяти, взаимодействие указателей и участков динамической памяти.

16. Одномерные динамические массивы. Двумерные динамические массивы.

17. Динамические структуры данных: однонаправленные и двунаправленные списки, очередь и стек, бинарные деревья.

18. Понятие ресурсной эффективности алгоритмов посредством анализа асимптотических функций временной и емкостной сложности.

18. Классификация алгоритмов на основе функции временной сложности, общие методы оценки трудоемкости алгоритмов.

19. Рекурсия и рекурсивные алгоритмы.

20. Алгоритм перебора с возвратом.

21. Алгоритмы поиска в линейных структурах.

22. Алгоритмы хеширования данных.

23. Определение и виды деревьев поиска, приемы снижения трудоемкости поиска в древовидных структурах.

24. Алгоритмы поиска в двоичных упорядоченных, случайных и сбалансированных в высоту (АВЛ) деревьях. Примеры программной реализации бинарного дерева поиска и АВЛ-дерева.

25. Определение и классификация алгоритмов сортировок массивов.

26. Бинарная пирамидальная сортировка.

27. Сортировка слиянием.

28. Сортировка Шелла.

29. Сортировка Хоара.

30. Внутренняя сортировка. Внешняя сортировка.

Критерии оценивания:

- 50-100 баллов («зачет») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой; наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины; наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов («незачет») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

**Вопросы к экзамену, 3 семестр (очная форма обучения),
2 курс (заочная форма обучения)
по дисциплине Инженерное программирование**

1. История ЯП. Классификация и область применения ЯП.

2. Парадигмы программирования. Процедурные языки. Аппликативные языки.

3. Языки логического программирования. Объектно-ориентированные языки.
4. Стандартизация языков программирования.
5. Среда проектирования VisualStudio.NET. Типы данных.
6. Поточковый ввод\вывода данных printf() и scanf(). Библиотека stdio.h. Операторы ввода\вывода (cout и cin). Библиотека <iostream>. Команда using namespace std.
7. Операция увеличения ++ и уменьшения --: «префиксная» форма, и «постфиксная» форма. Проверка четности числа. Логические операции. Очистка экрана. Файла conio.h.
8. Программирование разветвляющихся алгоритмов. Условные операторы. Оператор выбора switch .
9. Функции библиотеки math.h. Линейный алгоритм.
10. Программирование циклических алгоритмов.
11. Программирование с использованием одномерных массивов. Заполнение матрицы случайными числами.
12. Объектно-ориентированное программирование. Объект. Класс.
13. Инкапсуляция.
14. Полиморфизм.
15. Секции private, protected, public. Указатель this.
16. Переопределение операций.
17. Потоки работы с файлами. Поток и is_open().
18. Передача параметров через консоль (int argc, char *argv[]). Режимы файла. Бинарные файлы.
19. Решение. Проект.
20. Директивы #include, #define, #undef, #ifdef, #else, #if, #pragma.
21. Макроимена. __FILE__ и __LINE__.
22. Конструктор. Инициализация переменных в конструкторе. Конструктор без параметров. Копирующий конструктор. Содержательный конструктор.
23. Константные ссылки и указатели. Константные методы классов. Перегрузка.
24. Понятие наследования. Виды наследования.
25. Множественное наследование.
26. Дружественные функции. Дружественные члены класса (методы). Дружественные классы.
27. Виртуальные методы. Виртуальная функция. Виртуальные классы.
28. Абстрактные классы.
29. Виды ошибок. Исключительные ситуации. Обработка исключительных ситуаций. Операторы try и catch.
30. Шаблоны методов. Шаблоны классов. Назначение шаблонов.

Критерии оценивания:

- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка удовлетворительно) – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка неудовлетворительно) – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

Расчетные задачи по дисциплине Инженерное программирование

Расчетная задача № 1

Определить количество положительных элементов, расположенных ниже побочной диагонали матрицы.

Расчетная задача № 2

Определить количество отрицательных элементов, расположенных выше главной диагонали матрицы.

Расчетная задача № 3

Определить сумму отрицательных элементов, расположенных выше побочной диагонали матрицы.

Расчетная задача № 4

Определить произведение положительных элементов, расположенных ниже главной диагонали матрицы.

Расчетная задача № 5

Определить сумму элементов, расположенных на главной диагонали матрицы, и произведение элементов, расположенных на побочной диагонали матрицы.

Расчетная задача № 6

Определить количество четных элементов, расположенных на главной и побочной диагоналях.

Расчетная задача № 7

Найти максимальный среди элементов, лежащих ниже побочной диагонали.

Расчетная задача № 8

Найти минимальный среди элементов, лежащих выше главной диагонали.

Расчетная задача № 9

Найти максимальный среди элементов, лежащих выше побочной диагонали.

Расчетная задача № 10

Найти минимальный среди элементов, лежащих ниже главной диагонали.

Расчетная задача № 11

Найти в каждой строке матрицы максимальный элемент.

Расчетная задача № 12

Найти в каждом столбце матрицы минимальный элемент.

Расчетная задача № 13

Найти сумму элементов, расположенных в четных (по номеру) строках матрицы.

Расчетная задача № 14

Найти произведение элементов, расположенных в нечетных (по номеру) столбцах матрицы.

Расчетная задача № 15

Подсчитать сумму четных элементов и произведение нечетных элементов матрицы.

Критерии оценки:

- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка удовлетворительно) – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка неудовлетворительно) – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

Лабораторные задания по дисциплине Инженерное программирование

Лабораторное задание № 1

Типы данных в языке C++. Подставляемые (встраиваемые) функции. Перегрузка функций. Рекурсивные функции.

Лабораторное задание № 2

Производные типы. Тип указатель: указатели на объекты. Тип указатель: указатели на функции. Решение задач с использованием указателей.

Лабораторное задание № 3

Массивы: одномерные массивы, задачи поиска, замены и перестановок элементов массива, задачи сортировок элементов массива. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива, задачи сортировок и перестановок в двумерных массивах.

Лабораторное задание № 4

Решение задач на структуры и указатели, объединения.

Лабораторное задание № 5

Стандартные файлы и функции по работе с ними. Обработка файлов в потоковом режиме. Прямой доступ к данным файла. Работа с элементами файлов. Решение задач на обработку файлов.

Лабораторное задание № 6

Решение задач на динамическое выделение памяти. Массивы. Обработка элементов массива. Диагонали. Передача массива в функцию.

Лабораторное задание № 7

Алгоритмы обработки данных. Рекурсия и рекурсивные алгоритмы. Алгоритм перебора с возвратом. Алгоритмы поиска в линейных структурах. Алгоритмы хеширования данных.

Лабораторное задание № 8

Алгоритмы поиска в двоичных упорядоченных, случайных и сбалансированных в высоту (АВЛ) деревьях, программная реализация бинарного дерева поиска и АВЛ-дерева.

Лабораторное задание № 9

Алгоритмы сортировки массивов. Внутренняя сортировка. Внешняя сортировка

Лабораторное задание № 10

Разработка программы по обработке различных типов данных.

Разработка консольного приложения по обработке логических выражений. Реализация перевода чисел из одной системы счисления в другую. Разработка программы по обработке звука в консоли.

Лабораторное задание № 11. «Основные понятия объектно-ориентированного программирования (ООП)» Программирование разветвляющегося алгоритма.

Разработка программного кода с циклическим алгоритмом.

Разработка программы-класса, содержащей атрибуты, методы класса. Написать программу нахождение произведения. Свойства класса – две переменные. Метод класса – функция нахождения произведения.

Написать программу нахождение суммы. Свойства класса – три переменные. Метод класса – функция нахождения суммы.

Лабораторное задание № 12.

Реализовать объединение нескольких проектов в одно решение (mix). Класс Sum использовать 3 раза в разных файлах исходного кода двух проектов.

Лабораторное задание № 13.

Разработка программы с конструктором с использованием шаблона. Инициализация свойств. Const метод класса. Разработка программы с конструктором без использования шаблона. Передача параметров в программу.

Лабораторное задание № 14.

Реализация программного кода, используя наследование классов. Разработка программы с множественным наследованием. Класс Z наследует класс X и класс Y.

Лабораторное задание № 15.

Разработка функций-друзей и обращение к ним.

Разработка классов-друзей и обращение к ним.

Лабораторное задание № 16.

Разработка программы, содержащей класс с виртуальным методом

Лабораторное задание № 17.

Использование рабочей области в программном коде.

Пример работы с вектором (<vector>).

Лабораторное задание № 18.

Разработка программы, сочетающей технологию командной строки WINDOWS с технологиями файловых потоков для подсчета количества символов в файлах, перечисленных в командной строке. Разработка программы, которая создает бинарный файл, содержащий информацию о товаре (используя структуру): наименование, цена, вес, применяя манипуляторы для форматирования экранного вывода информации из файла.

2. Методические рекомендации по выполнению лабораторных заданий

Лабораторные задания выполняются с учетом приобретенных знаний по предшествующим дисциплинам, теоретического материала дисциплины, с помощью и консультациями (при необходимости) преподавателя на занятиях.

3. Критерии оценки:

- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению

полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка удовлетворительно) – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка неудовлетворительно) – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

Индивидуальное задание (очная форма обучения) Контрольная работа (заочная форма обучения) по дисциплине Инженерное программирование

Целью индивидуального задания (контрольной работы) является практическая реализация освоенных принципов разработки этапов жизненного цикла программы от модели анализа: информационной, функциональной, поведенческой к абстракциям архитектурного и алгоритмического уровней, а также приобретение обучающимися навыков тестирования, отладки и документирования разработанных проектов.

Типовая тематика индивидуального задания (контрольной работы):

1. «Разработка проекта (программного модуля) автоматизации расписания занятий в университете»: В системе должны поддерживаться режимы поиска занятия по заданному критерию (время, преподаватель), регистрации занятий, учета занятий по типу, генерации расписаний.

2. «Разработка проекта (программного модуля) автоматизации библиотеки»: В системе должны поддерживаться режимы поиска книги по заданному критерию (автор, название), заказа книги, учета клиентов и книг в книгохранилище, выдачи отчетов по запросам (местонахождение книги в архиве или ее отсутствие), выдачи документов о должниках.

3. «Разработка проекта (программного модуля) автоматизации школы»: В системе должны поддерживаться режимы учета классов и учеников в них, регистрации нового ученика, учета посещаемости занятий и оценок учащихся, генерации отчетов по успеваемости учеников.

4. «Разработка проекта (программного модуля) поликлиники»: В системе должны поддерживаться режимы учета докторов и пациентов, поиска пациента по заданному критерию, анализа заболеваемости по районам, по месяцам и выдачи справок о болезни.

5. «Разработка проекта (программного модуля) автоматизации аптеки»: В системе должны поддерживаться режимы поиска лекарства по заданному критерию (название, болезнь, цена), заказа, покупки и учета лекарств, анализа спроса на лекарства в зависимости от стоимости, времени года и т. п., выдачи соответствующих отчетов.

6. «Разработка проекта (программного модуля) автоматизации музея»: В системе должны поддерживаться режимы учета посетителей и экспонатов музея, регистрации новых экспонатов, поиска экспонатов по заданному критерию (название, эпоха и т. д.), учета доходов и расходов, связанных с проведением экскурсий, выдачи отчетов по запросам.

7. Создание системы обмена сообщениями через протокол TCP.

8. Создание системы обмена изображениями (видео) через протокол UDP.

9. Создание FTP клиента.

10. Применение современных ИТ в работе коммивояжера (администратора, риэлтора, менеджера по закупкам, менеджера по продажам, менеджера по услугам, товароведа, в работе электронного бизнеса и т.д.).

2. Методические рекомендации по выполнению индивидуального задания
Индивидуальное задание (контрольная работа) выполняется с учетом приобретенных знаний по предшествующим дисциплинам, теоретического материала дисциплины.

Основными этапами выполнения индивидуального задания являются:

1. Постановка задачи.
2. Построение модели.
3. Разработка алгоритма.
4. Реализация алгоритма.
5. Проверка программы.
6. Составление отчета.

Постановка задачи. Это начальный этап и начинается он с ознакомления задачей и рекомендуемой литературой. Прежде чем решать задачу, необходимо ее точно сформулировать. Процесс точной формулировки задачи сводится к постановке правильных вопросов:

Понятна ли терминология, используемая в предварительной формулировке?

Что дано? Что нужно найти?

Как определить решение?

Каких данных не хватает и все ли они нужны?

Являются ли какие-то имеющиеся данные бесполезными? Какие сделаны допущения?

Возможны и другие вопросы в зависимости от конкретной задачи. Точную постановку задачи необходимо сформулировать в процессе консультации с преподавателем.

Построение модели. Задача четко поставлена, нужно сформулировать для нее математическую модель. Это очень важный шаг в процессе решения, и его надо хорошо обдумать. Выбор модели существенно влияет на остальные этапы в процессе решения. Большинство задач должно рассматриваться индивидуально.

Приступая к разработке модели, следует задать по крайней мере несколько основных вопросов:

Существует ли математическая величина, ассоциируемая с искомым результатом?

Какие математические структуры больше всего подходят для задачи?

Имеются ли какие-нибудь полезные отношения между объектами модели?

Существуют ли решенные аналогичные задачи?

Большинство решаемых задач, как правило, являются модификациями ранее решенных и для продвижения вперед приходится руководствоваться накопленным опытом.

Сделав выбор математической структуры, задачу следует переформулировать в терминах соответствующих математических объектов.

Разработка алгоритма. Как только задача четко поставлена и для нее построена модель, необходимо приступить к разработке алгоритма ее решения. Выбор метода разработки, сильно зависящий от выбора модели, может в значительной степени повлиять на эффективность алгоритма решения. Два разных алгоритма могут быть правильными, но очень сильно отличаться по эффективности. Доказательство правильности алгоритма — это один из наиболее трудных этапов создания алгоритма. Вероятно, наиболее распространенная процедура доказательства правильности программы — это прогон ее на разных тестах. Если выданные программой ответы могут быть подтверждены известными или вычисленными вручную данными, возникает вывод, что программа «работает». Однако этот метод редко исключает все сомнения; может существовать случай, в котором программа «не работает».

Можно предложить следующую общую методику доказательства правильности алгоритма. Предположим, что алгоритм описан в виде последовательности шагов, скажем, от

шага 0 до шага n. Необходимо предложить некое обоснование правомерности для каждого шага. В частности, может потребоваться лемма об условиях, действующих до и после пройденного шага. Затем необходимо предложить доказательство конечности алгоритма, при этом будут проверены все подходящие входные данные и получены все подходящие выходные данные.

Реализация алгоритма. Необходимо построить целую систему структур данных (классов) для представления важных аспектов используемой модели. Руководствуясь технологией ООП программа должна быть представлена совокупностью взаимодействующих объектов. Следует внимательно рассмотреть приведенные выше примеры с тем, чтобы попытаться применить разработанные в них классы в качестве прототипов для разработки путем наследования собственных объектов. При этом желательно исключить использование многих глобальных переменных и строгую локализацию данных и действий рамками отдельных объектов, что является основой высокой надежности программы.

Проверка программы. Проверка программы может быть охарактеризована как экспериментальное подтверждение того факта, что программа делает именно то, что должна делать. Проверка программы является также экспериментальной попыткой установить границы использования алгоритма (проекта). Недостаточно доказать правильность алгоритма. Окончательная программа должна быть тщательно проверена и оттестирована. Как выбрать входные данные для тестирования? На этот вопрос невозможно дать общего ответа. Для любого алгоритма ответ зависит от сложности программы, имеющегося ресурса времени, а также от числа вводов (т. е. вариантов входных данных), для которых можно установить правильность выводов, и т.д. Обычно множество всех вводов огромно, и полная проверка практически невозможна. Необходимо выбрать множество вводов, которые проверяют каждый участок программы.

Составление текстовой части индивидуального задания. Индивидуальное задание должен включать следующие разделы.

1. Содержание, включающее наименование всех разделов и пунктов с указанием номеров страниц.

2. Введение.

2.1 Дается характеристика предметной области, к которой относится решаемая задача и обосновывается ее актуальность.

2.2 Цель работы.

Формулируется цель выполнения индивидуального задания.

2.3 Постановка задач.

В этом разделе требуется формализовать задачи, указать возможные ограничения на их решение, CASE-средства, ИТ-технологии и т.п.

3. Техническое задание.

В соответствии с вариантом формулируется задание по индивидуальному заданию.

4. Теоретическая часть, освещающую теоретические аспекты темы;

5. Практическая часть, в которой разработка программы и ее результаты.

5.1 Анализ и разработка проекта (основной раздел отчета).

Должен отражать результаты анализа возможных вариантов решения задачи и выбора среди них наиболее рационального. Приводятся математические выкладки и рисунки, поясняющие зависимости параметров решения задачи от данных. Обосновывается выбор структур данных и основных операций над ними. Приводятся результаты моделирования программ-прототипов (если они использовались для разработки алгоритма). Определяются источники и форматы исходных данных и содержание вывода программы.

5.2 Алгоритм работы проекта.

В этом разделе приводится блок-схема алгоритма программы.

5.3 Текст программы.

Приводится текст программы с комментариями.

5.4 Результаты моделирования программы на ПК.

Должны быть приведены результаты тестирования программы с различными наборами данных, в том числе и с некорректными данными.

6. Выводы.

Приводятся комментарии к результатам и рекомендации к дальнейшему совершенствованию программы.

7. Использованная литература.

8. Приложения

Включают материалы иллюстративного и вспомогательного характера (таблицы большого формата; дополнительные расчеты; распечатки и проч.) Приложения обозначаются русскими заглавными буквами - А, Б, В и т.д. (например, «Приложение А»), располагаются в виде заголовка, по центру.

Таблицы, рисунки, формулы оформляются в соответствии с внутривузовским изданием для нормоконтроля. На все таблицы, рисунки, литературные источники, приложения в тексте должны быть ссылки.

Оформление индивидуального задания (контрольной работы) должно соответствовать требованиям государственных стандартов, в т.ч. и методических рекомендаций вуза (кафедры). Текст работы должен быть набран на белой бумаге формата А4 с одной стороны листа. Размер шрифта: 12-14, интервал: 1,5. Поля: левое – 30 мм, правое – 10 мм, верхнее – 20 мм, нижнее – 20 мм.

3. Критерии оценки контрольной работы:

- 84-100 баллов (оценка «отлично») – разработанный интерфейс программы и ее функциональные возможности соответствуют требованиям задания контрольной работы; текстовое описание контрольной работы составлено в полном объеме и аккуратно; программа работает без сбоев для всех типовых задач, для которых она разрабатывалась, предусмотрена защита от ввода некорректных данных; во время защиты обучающийся показал свободное владение темой проекта, знание используемых компонентов, их свойств и специальных функций; изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и

логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – разработанный интерфейс программы и ее функциональные возможности соответствуют требованиям задания контрольной работы; текстовое описание контрольной работы составлено в не полном объеме; программа работает без сбоев для всех типовых задач, для которых она разрабатывалась, не предусмотрена защита от ввода некорректных данных; во время защиты студент показал свободное владение темой проекта, знание используемых компонентов, их свойств и специальных функций; наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка удовлетворительно) – разработанный интерфейс программы и ее функциональные возможности соответствуют требованиям задания контрольной работы; текстовое описание контрольной работы составлено не в полном объеме и не достаточно аккуратно; программа работает без сбоев, не для всех типовых задач, для которых она разрабатывалась, не предусмотрена защита от ввода некорректных данных; во время защиты студент показал слабые знания по теме проекта; наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка неудовлетворительно) – разработанный интерфейс программы и ее функциональные возможности не соответствуют требованиям задания контрольной работы; текстовое описание контрольной работы составлено не в полном объеме и не аккуратно; программа работает со сбоями, не для всех типовых задач, для которых она разрабатывалась, не предусмотрена защита от ввода некорректных данных; во время защиты студент показал слабые знания по теме проекта; ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

4. Методические материалы, определяющие процедуры оценивания результатов освоения образовательной программы

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

Текущий контроль успеваемости проводится с использованием оценочных средств, представленных в п. 3 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

Промежуточная аттестация проводится в форме зачета, экзамена.

Зачет проводится по окончании теоретического обучения до начала экзаменационной сессии.

Экзамен проводится по расписанию экзаменационной сессии в устном виде. Количество вопросов в экзаменационном задании – 3. Объявление результатов производится в день экзамена.

Результаты аттестации заносятся в экзаменационную ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику сессии, должны ликвидировать задолженность в установленном порядке.

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ростовский государственный экономический университет (РИНХ)»

Рассмотрено и одобрено
на заседании кафедры Информационные технологии и
защита информации
Протокол № 10 от 11.05.2018 г.
Зав.кафедрой _____ Тищенко Е.Н.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Инженерное программирование

Направление подготовки

09.03.04 Программная инженерия

Уровень образования

Бакалавриат

Составитель

Жилина Е.В., доцент, к.э.н.
(подпись) Ф.И.О., должность, ученая степень, ученое звание

Ростов-на-Дону, 2018

Методические указания по освоению дисциплины «Инженерное программирование» адресованы студентам всех форм обучения.

Учебным планом по направлению подготовки 09.03.04 «Программная инженерия предусмотрены следующие виды занятий:

- лекционные
- лабораторные

В ходе лекционных занятий рассматриваются основные теоретические вопросы, даются рекомендации для самостоятельной работы и подготовке к лабораторным занятиям.

При подготовке к лабораторным занятиям каждый студент должен:

- изучить рекомендованную учебную литературу;
- изучить конспекты лекций;
- подготовить ответы на все вопросы по изучаемой теме;
- письменно решить домашнее задание, рекомендованные преподавателем при изучении каждой темы.

По согласованию с преподавателем студент может подготовить реферат, доклад или сообщение по теме занятия. В процессе подготовки к лабораторным занятиям студенты могут воспользоваться консультациями преподавателя.

Вопросы, не рассмотренные на аудиторных занятиях, должны быть изучены студентами в ходе самостоятельной работы. Контроль самостоятельной работы студентов над учебной программой курса осуществляется в ходе занятий методом устного опроса или контрольной работы. В ходе самостоятельной работы каждый студент обязан прочитать основную и по возможности дополнительную литературу по изучаемой теме, дополнить конспекты недостающим материалом, выписками из рекомендованных первоисточников. Выделить непонятные термины, найти их значение в энциклопедических словарях.

Студент должен готовиться к предстоящим лабораторным занятиям по всем, обозначенным в рабочей программе дисциплины вопросам.

При реализации различных видов учебной работы используются разнообразные (в т.ч. интерактивные) методы обучения, в частности:

- интерактивная доска для подготовки и проведения лекционных занятий.

Для подготовки к занятиям, текущему контролю и промежуточной аттестации студенты могут воспользоваться электронной библиотекой ВУЗа <http://library.rsue.ru/>. Также обучающиеся могут взять на дом необходимую литературу на абонементе вузовской библиотеки или воспользоваться читальными залами вуза.