

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 15.11.2024 13:57:35

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

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

УТВЕРЖДАЮ

Начальник

учебно-методического управления

Платонова Т.К.

«25» июня 2024 г.

**Рабочая программа дисциплины  
Архитектура вычислительных систем**

Направление 09.03.04 "Программная инженерия"

Направленность 09.03.04.01 Системное и прикладное программное обеспечение

Для набора 2022 года

Квалификация  
Бакалавр

**КАФЕДРА Информационные технологии и программирование****Распределение часов дисциплины по семестрам**

Семестр (<Курс>.<Семестр на курсе>)	8 (4.2)		Итого	
	7			
Неделя	7			
Вид занятий	УП	РП	УП	РП
Лекции	14	14	14	14
Лабораторные	14	14	14	14
Итого ауд.	28	28	28	28
Контактная работа	28	28	28	28
Сам. работа	80	80	80	80
Итого	108	108	108	108

**ОСНОВАНИЕ**

Учебный план утвержден учёным советом вуза от 25.06.2024 г. протокол № 18.

Программу составил(и): к.ф.-м.н., доцент, Карнаухов С.Н.

Зав. кафедрой: Ефимова Е.В.

Методический совет направления: д.э.н., профессор Тищенко Е.Н.

### 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	формирование и закрепление системного подхода к изучению структуры и принципах работы вычислительных систем разного назначения.
-----	---

### 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**ОПК-3: способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности**

#### В результате освоения дисциплины обучающийся должен:

**Знать:**

- основные принципы построения ЭВМ и архитектуру вычислительных систем (соотнесено с индикатором ОПК-3.1);

**Уметь:**

-определять оптимальную конфигурацию программного оборудования и характеристики аппаратных устройств для решения практических задач (соотнесено с индикатором ОПК-3.2)

**Владеть:**

- навыками практического использования свойств архитектуры вычислительной системы, в рамках поставленной задачи (соотнесено с индикатором ОПК-3.3)

### 3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

#### Раздел 1. Введение в АВС

№	Наименование темы / Вид занятия	Семестр / Курс	Часов	Компетенции	Литература
1.1	Введение в вычислительные системы.История развития. Структура вычислительных систем. Основные этапы развития вычислительных систем.Вычислительные системы: понятие и назначение. Структура стандартов ИСО. Классы программных продуктов. Требования к ВС / Лек /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
1.2	Структура стандартов ИСО. Анализ структуры ИСО. Классификация программных продуктов. основные требования, предъявляемые к вычислительным системам. / Лаб /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
1.3	Архитектура вычислительных систем: уровни вычислительных систем. Содержание архитектуры вычислительных систем, основные уровни. / Лек /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
1.4	Архитектура вычислительных систем: архитектура статически типизированного универсального языка программирования, ориентированная на процедурный подход / Лаб /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3
1.5	архитектура статически типизированного универсального языка программирования, ориентированная на объектно-ориентированный подход; / Лаб /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
1.6	Архитектура и принципы работы персонального компьютера; физическую сущность работы, назначение, виды и характеристики;основных устройств компьютера;основы машинных языков;структуру и функционирование операционных систем;организацию и принципы функционирования компьютерных сетей;уровни, протоколы, технологии эталонной модели открытых систем OSI;основы сетевых операционных систем; тенденции развития аппаратного и программного обеспечения вычислительных и коммуникационных систем / Ср /	8	30	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
<b>Раздел 2. Реализация концепций управления производством в ВС</b>					
№	Наименование темы / Вид занятия	Семестр / Курс	Часов	Компетенции	Литература

2.1	архитектура ВС, ориентированная на динамическую типизацию и процедурное программирование / Лаб /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
2.2	Методология MRP, MRP II и ERP»: обзор стандартов, методологии, модули систем / Лек /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
2.3	Функциональность систем MRP». Анализ входных и выходных параметров процессов / Лек /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
2.4	архитектура ВС машинного уровня с программированием на языке ассемблера / Лаб /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
2.5	Функциональность систем CRP». Входные и выходные параметры процессов. Процесс планирования / Лек /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
2.6	архитектура ВС с поддержкой параллельного многопоточного программирования с использованием потоков posix threads / Лаб /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
2.7	Классификация вычислительных систем. Особенности и ограничения в эксплуатации. / Лек /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3
2.8	Виды сервисов в сети Интернет. Адресация компьютеров в сети Интернет. Принцип построения и работы электронной почты в сети Интернет. Сервис WWW. Поиск информации в сети Интернет. Принцип построения корпоративных компьютерных сетей. Многоядерные структуры микропроцессоров. Принцип построения параллельных вычислений при использовании многоядерных микропроцессоров. / Ср /	8	25	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4

### Раздел 3. Этапы разработки ВС

№	Наименование темы / Вид занятия	Семестр / Курс	Часов	Компетенции	Литература
3.1	Классический жизненный цикл»: жизненный цикл, инженерный цикл. Макетирование: формы, последовательность действий / Лаб /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
3.2	жизненный цикл, инженерный цикл. Макетирование: формы, последовательность действий / Лек /	8	2	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4
3.3	Структура семейства стандартов ИСО 9000. Макетирование: формы, последовательность действий Реализация концепций управления производством в ВС. Классы программных продуктов Требования к ВС комплексные системы управления предприятием Объединенная система планирования MRP-CRP / Ср /	8	25	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3
3.4	/ Зачёт /	8	0	ОПК-3	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3, Л2.4

### 4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Структура и содержание фонда оценочных средств для проведения текущей и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

### 5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

#### 5.1. Основная литература

	Авторы,	Заглавие	Издательство, год	Колич-во
Л1.1	Лиманова, Н. И.	Архитектура вычислительных систем и компьютерных сетей: учебное пособие	Самара: Поволжский государственный университет телекоммуникаций и информатики, 2017	<a href="https://www.iprbookshop.ru/75368.html">https://www.iprbookshop.ru/75368.html</a> неограниченный доступ для зарегистрированных пользователей
Л1.2	Кравченко П. П., Стулин Е. В., Хусаинов Н. Ш.	Моделирование вычислительных систем обработки запросов на языке GPSS WORLD: учебное пособие по курсу "Архитектура вычислительных систем": учебное пособие	Таганрог: Южный федеральный университет, 2016	<a href="https://biblioclub.ru/index.php?page=book&amp;id=493201">https://biblioclub.ru/index.php?page=book&amp;id=493201</a> неограниченный доступ для зарегистрированных пользователей

#### 5.2. Дополнительная литература

	Авторы,	Заглавие	Издательство, год	Колич-во
Л2.1	Бедердинова О. И., Водовозова Ю. А.	Информационные технологии общего назначения: учебное пособие	Архангельск: Северный (Арктический) федеральный университет, 2015	<a href="https://biblioclub.ru/index.php?page=book&amp;id=436288">https://biblioclub.ru/index.php?page=book&amp;id=436288</a> неограниченный доступ для зарегистрированных пользователей
Л2.2	Гриценко Ю. Б.	Вычислительные системы, сети и телекоммуникации: учебное пособие	Томск: ТУСУ, 2015	<a href="https://biblioclub.ru/index.php?page=book&amp;id=480639">https://biblioclub.ru/index.php?page=book&amp;id=480639</a> неограниченный доступ для зарегистрированных пользователей
Л2.3	Городничев, М. Г.	Учебно-методическое пособие по дисциплине Архитектура вычислительных систем	Москва: Московский технический университет связи и информатики, 2016	<a href="https://www.iprbookshop.ru/61466.html">https://www.iprbookshop.ru/61466.html</a> неограниченный доступ для зарегистрированных пользователей
Л2.4		БИТ. Бизнес & Информационные технологии: журнал	Москва: Положевец и партнеры, 2019	<a href="https://biblioclub.ru/index.php?page=book&amp;id=562412">https://biblioclub.ru/index.php?page=book&amp;id=562412</a> неограниченный доступ для зарегистрированных пользователей

### 5.3 Профессиональные базы данных и информационные справочные системы

Консультант +  
Гарант <https://www.garant.ru/>  
Бесплатная база данных ГОСТ. <https://docplan.ru>

### 5.4. Перечень программного обеспечения

Операционная система РЕД ОС  
ОС Linux  
Ассемблер

### 5.5. Учебно-методические материалы для студентов с ограниченными возможностями здоровья

При необходимости по заявлению обучающегося с ограниченными возможностями здоровья учебно-методические материалы предоставляются в формах, адаптированных к ограничениям здоровья и восприятия информации. Для лиц с нарушениями зрения: в форме аудиофайла; в печатной форме увеличенным шрифтом. Для лиц с нарушениями слуха: в форме электронного документа; в печатной форме. Для лиц с нарушениями опорно-двигательного аппарата: в форме электронного документа; в печатной форме.

## 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Помещения для всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения:

- столы, стулья;
- персональный компьютер / ноутбук (переносной);
- проектор;
- экран / интерактивная доска.

Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными программными средствами и выходом в Интернет.

## 7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

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

1.1 Критерии оценивания компетенций

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
ОПК-3 способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности			
З. основные принципы построения ЭВМ и архитектуру вычислительных систем	изучает основные принципы и архитектуру вс при подготовке к зачету и опросу	полнота и содержательность ответа на опросе и зачете, умение приводить примеры	З (1-28) О (1-25)
У. определять оптимальную конфигурацию программного оборудования и характеристики аппаратных устройств для решения практических задач	Определяет конфигурацию для построения вс при выполнении лабораторных и практико-ориентированных заданий	Правильность выполнения лабораторных и практико-ориентированных заданий	ПОЗЗ (1-5) ЛР (1-5)
В. навыками практического использования свойств архитектуры вычислительной системы, в рамках поставленной задачи	Выполняет выбор технологий для проектирования вс при выполнении лабораторных и практико-ориентированных заданий	Объем и правильность выполнения лабораторных и практико-ориентированных заданий	ПОЗЗ (1-5) ЛР (1-5)

О – опрос; З – вопросы к зачету; ПОЗЗ - практико-ориентированные задания к зачету; ЛР –лабораторные задания;

1.2 Шкалы оценивания:

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

50-100 баллов (зачет)

0-49 баллов (незачет)

**2. 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**

**Вопросы к зачету**

1. Основные понятия вычислительных систем.
2. Структура стандартов ИСО.
3. Классы программных продуктов.
4. Требования к ВС.
5. Архитектура ВС.
6. Уровни ВС.
7. История развития.
8. Структура ВС.
9. Методология MRP
10. Методология MRP II
11. Методология ERP.
12. Функциональность систем MRP.
13. Входные и выходные параметры процессов.
14. Функциональность систем CRP.
15. Процесс планирования.

16. Классификация ВС.
17. Классический жизненный цикл
18. Инженерный цикл.
19. Макетирование: формы, последовательность действий.
20. Стратегия разработки ПО.
21. Многоуровневая организация архитектур ВС
22. Сравнительный анализ особенностей архитектур компьютеров различных уровней
23. Практическое использование методов программирования, учитывающих архитектурные особенности.
24. Архитектура статически типизированного универсального языка программирования, ориентированная на процедурный подход
25. Архитектура статически типизированного универсального языка программирования, ориентированная на объектно-ориентированный подход
26. Архитектура ВС, ориентированная на динамическую типизацию и процедурное программирование
27. Архитектура ВС машинного уровня с программированием на языке ассемблера
28. Архитектура ВС с поддержкой параллельного многопоточного программирования с использованием потоков posix threads.

### **Практико-ориентированные задания к зачету**

В ходе выполнения задания необходимо написать программу, которая должна быть оформлена в виде консольного приложения. Архитектуры вычислительной сети:

- 1) архитектура статически типизированного универсального языка программирования, ориентированная на процедурный подход: Упорядочить элементы контейнера по возрастанию используя сортировку с помощью прямого включения (Straight Insertion). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.
- 2) архитектура статически типизированного универсального языка программирования, ориентированная на объектно-ориентированный подход: Упорядочить элементы контейнера по возрастанию используя сортировку с помощью «дерева» (Heap Sort). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив
- 3) архитектура ВС, ориентированная на динамическую типизацию и процедурное программирование: Упорядочить элементы контейнера по убыванию используя сортировку Сортировка с помощью прямого выбора (Straight Selection). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив
- 4) архитектура ВС машинного уровня с программированием на языке ассемблера: Удалить из контейнера те элементы, для которых значение, полученное с использованием функции, общей для всех альтернатив, больше чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции. Остальные элементы передвинуть в начало контейнера с сохранением порядка
- 5) архитектура ВС с поддержкой параллельного многопоточного программирования с использованием потоков posix threads: Переместить в начало контейнера те элементы, для которых значение, полученное с использованием функции, общей для всех альтернатив, меньше чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции. Остальные элементы сдвинуть к началу без изменения их порядка.

### **Критерии оценивания:**

50-100 баллов (зачет) – изложенный материал верен, наличие знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

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

## Опрос

1. Требования к ВС.
2. Архитектура ВС.
3. Уровни ВС.
4. История развития.
5. Структура ВС.
6. Методология MRP
7. Методология MRP II
8. Методология ERP.
9. Функциональность систем MRP.
10. Входные и выходные параметры процессов.
11. Функциональность систем CRP.
12. Процесс планирования.
13. Классификация ВС.
14. Классический жизненный цикл
15. Инженерный цикл.
16. Макетирование: формы, последовательность действий.
17. Стратегия разработки ПО.
18. Многоуровневая организация архитектур ВС
19. Сравнительный анализ особенностей архитектур компьютеров различных уровней
20. Практическое использование методов программирования, учитывающих архитектурные особенности.
21. Архитектура статически типизированного универсального языка программирования, ориентированная на процедурный подход
22. Архитектура статически типизированного универсального языка программирования, ориентированная на объектно-ориентированный подход
23. Архитектура ВС, ориентированная на динамическую типизацию и процедурное программирование
24. Архитектура ВС машинного уровня с программированием на языке ассемблера
25. Архитектура ВС с поддержкой параллельного многопоточного программирования с использованием потоков posix threads..

### Критерии оценивания:

- 1 балл выставляется обучающемуся, если изложенный материал фактически верен и логически обоснован.

- 0 баллов ответ неверный.

Максимальное количество баллов: 25 баллов.

## Лабораторные задания

### Лабораторное задание №1

В ходе выполнения задания необходимо написать программу, которая должна быть оформлена в виде консольного приложения. Вычислительную систему с архитектурой x86-64; операционную систему Linux; Архитектуры вычислительной сети:

архитектура статически типизированного универсального языка программирования, ориентированная на процедурный подход: Упорядочить элементы контейнера по возрастанию используя сортировку с помощью прямого включения (Straight Insertion). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

Провести отладку и тестирование разработанной программы на заранее подготовленных тестовых наборах данных. Количество тестовых наборов данных – не менее пяти. Число уникальных элементов в тестовых наборах должно варьироваться от нуля до 10000. При необходимости, программа должна правильно обрабатывать переполнение по данным. Тестовые наборы до 20 элементов должны вводиться из заранее подготовленных тестовых файлов. Тестовые данные с большим числом элементов должны порождаться программно с использованием генераторов случайных наборов данных. Данные формируемые генератором случайных наборов данных должны поддерживать допустимые значения. Управление вводом данных задается из командной строки



Описать структуру используемой ВС с наложением на нее обобщенной схемы разработанной программы

### **Лабораторное задание №2**

В ходе выполнения задания необходимо написать программу, которая должна быть оформлена в виде консольного приложения. Вычислительную систему с архитектурой x86-64; операционную систему Linux; Архитектуры вычислительной сети:

архитектура статически типизированного универсального языка программирования, ориентированная на объектно-ориентированный подход: Упорядочить элементы контейнера по возрастанию используя сортировку с помощью «дерева» (Heap Sort). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив

Провести отладку и тестирование разработанной программы на заранее подготовленных тестовых наборах данных. Количество тестовых наборов данных – не менее пяти. Число уникальных элементов в тестовых наборах должно варьироваться от нуля до 10000. При необходимости, программа должна правильно обрабатывать переполнение по данным. Тестовые наборы до 20 элементов должны вводиться из заранее подготовленных тестовых файлов. Тестовые данные с большим числом элементов должны порождаться программно с использованием генераторов случайных наборов данных. Данные формируемые генератором случайных наборов должны поддерживать допустимые значения. Управление вводом данных задается из командной строки

Описать структуру используемой ВС с наложением на нее обобщенной схемы разработанной программы

### **Лабораторное задание №3**

В ходе выполнения задания необходимо написать программу, которая должна быть оформлена в виде консольного приложения. Вычислительную систему с архитектурой x86-64; операционную систему Linux; Архитектуры вычислительной сети:

архитектура ВС, ориентированная на динамическую типизацию и процедурное программирование: Упорядочить элементы контейнера по убыванию используя сортировку Сортировка с помощью прямого выбора (Straight Selection). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив

Провести отладку и тестирование разработанной программы на заранее подготовленных тестовых наборах данных. Количество тестовых наборов данных – не менее пяти. Число уникальных элементов в тестовых наборах должно варьироваться от нуля до 10000. При необходимости, программа должна правильно обрабатывать переполнение по данным. Тестовые наборы до 20 элементов должны вводиться из заранее подготовленных тестовых файлов. Тестовые данные с большим числом элементов должны порождаться программно с использованием генераторов случайных наборов данных. Данные формируемые генератором случайных наборов должны поддерживать допустимые значения. Управление вводом данных задается из командной строки

### **Лабораторное задание №4**

В ходе выполнения задания необходимо написать программу, которая должна быть оформлена в виде консольного приложения. Вычислительную систему с архитектурой x86-64; операционную систему Linux; Архитектуры вычислительной сети:

архитектура ВС машинного уровня с программированием на языке ассемблера: Удалить из контейнера те элементы, для которых значение, полученное с использованием функции, общей для всех альтернатив, больше чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции. Остальные элементы передвинуть в начало контейнера с сохранением порядка.

Провести отладку и тестирование разработанной программы на заранее подготовленных тестовых наборах данных. Количество тестовых наборов данных – не менее пяти. Число уникальных элементов в тестовых наборах должно варьироваться от нуля до 10000. При необходимости, программа должна правильно обрабатывать переполнение по данным. Тестовые наборы до 20 элементов должны вводиться из заранее подготовленных тестовых файлов. Тестовые данные с большим числом элементов должны порождаться программно с использованием генераторов случайных наборов

данных. Данные формируемые генератором случайных наборов должны поддерживать допустимые значения. Управление вводом данных задается из командной строки

### **Лабораторное задание №5**

В ходе выполнения задания необходимо написать программу, которая должна быть оформлена в виде консольного приложения. Вычислительную систему с архитектурой x86-64; операционную систему Linux; Архитектуры вычислительной сети:

архитектура ВС с поддержкой параллельного многопоточного программирования с использованием потоков posix threads: Переместить в начало контейнера те элементы, для которых значение, полученное с использованием функции, общей для всех альтернатив, меньше чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции. Остальные элементы сдвинуть к началу без изменения их порядка.

Провести отладку и тестирование разработанной программы на заранее подготовленных тестовых наборах данных. Количество тестовых наборов данных – не менее пяти. Число уникальных элементов в тестовых наборах должно варьироваться от нуля до 10000. При необходимости, программа должна правильно обрабатывать переполнение по данным. Тестовые наборы до 20 элементов должны вводиться из заранее подготовленных тестовых файлов. Тестовые данные с большим числом элементов должны порождаться программно с использованием генераторов случайных наборов данных. Данные формируемые генератором случайных наборов должны поддерживать допустимые значения. Управление вводом данных задается из командной строки

### **Критерии оценивания:**

15 б. – задание выполнено верно;

14-10 б. – при выполнении задания были допущены неточности, не влияющие на результат;

9-5 б. – при выполнении задания были допущены ошибки;

4-1 б. – при выполнении задания были допущены существенные ошибки;

0 б. – задание не выполнено.

Максимальное количество баллов- 75.

## **3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

**Текущий контроль** успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

**Промежуточная аттестация** проводится в форме зачета.

Количество вопросов в задании – 3: два теоретических вопроса и одно практико-ориентированное задание. Объявление результатов производится в день зачета. Результаты аттестации заносятся в ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику, должны ликвидировать задолженность в установленном порядке.

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учебным планом предусмотрены следующие виды занятий:

- лекции;
- лабораторные занятия.

В ходе лекционных занятий рассматриваются теоретические вопросы с учетом практико-ориентированности изучаемой дисциплины, даются рекомендации для самостоятельной работы и подготовке к лабораторным занятиям.

В ходе лабораторных занятий углубляются и закрепляются знания студентов по ряду рассмотренных на лекциях вопросов, развиваются навыки выбора методов программирования, учитывающих архитектурные особенности.

При подготовке к лабораторным занятиям каждый обучающийся должен:

- изучить рекомендованную учебную литературу;
- изучить практические примеры, рассмотренные на лекциях.

В процессе подготовки к лабораторным занятиям студенты могут воспользоваться консультациями преподавателя.

Вопросы, не рассмотренные на лекциях и лабораторных занятиях, должны быть изучены студентами в ходе самостоятельной работы. Контроль самостоятельной работы студентов над учебной программой курса осуществляется в ходе занятий методом опроса или при выполнении лабораторных заданий с учетом индивидуальности представленного решения. В ходе самостоятельной работы каждый студент обязан прочитать основную и по возможности дополнительную литературу по изучаемой теме, дополнить конспекты лекций недостающим материалом, выписками из рекомендованных первоисточников. Выделить непонятные термины, найти их значение в энциклопедических словарях.

Для подготовки к занятиям, текущему контролю и промежуточной аттестации студенты могут воспользоваться электронно-библиотечными системами. Также обучающиеся могут взять на дом необходимую литературу на абонементе университетской библиотеки или воспользоваться читальными залами.