

Документ подписан простой электронной подписью
 Информация о владельце:
 ФИО: Гаранин Максим Александрович
 Должность: Ректор
 Дата подписания: 08.10.2025 14:50:46
 Уникальный программный ключ:
 7708e3a47e66a8ee02711b298d7c78bd1e40bf88

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПРИВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ»

Системное программирование

рабочая программа дисциплины (модуля)

Направление подготовки 09.03.02 Информационные системы и технологии
 Направленность (профиль) Информационные системы и технологии на транспорте

Квалификация **бакалавр**
 Форма обучения **очная**
 Общая трудоемкость **3 ЗЕТ**

Виды контроля в семестрах:
 зачеты 5

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	5 (3.1)		Итого	
Неделя	16 1/6			
Вид занятий	уп	рп	уп	рп
Лекции	16	16	16	16
Лабораторные	32	32	32	32
Конт. ч. на аттест. в период ЭС	0,15	0,15	0,15	0,15
В том числе в форме практ.подготовки	32	32	32	32
Итого ауд.	48	48	48	48
Контактная работа	48,15	48,15	48,15	48,15
Сам. работа	51	51	51	51
Часы на контроль	8,85	8,85	8,85	8,85
Итого	108	108	108	108

Программу составил(и):

к.т.н., доцент, Фатеев В.А.

Рабочая программа дисциплины

Системное программирование

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.02 Информационные системы и технологии (приказ Минобрнауки России от 19.09.2017 г. № 926)

составлена на основании учебного плана: 09.03.02-25-2-ИСТб.plm.plx

Направление подготовки 09.03.02 Информационные системы и технологии Направленность (профиль) Информационные системы и технологии на транспорте

Рабочая программа одобрена на заседании кафедры

Цифровые технологии

Зав. кафедрой к.т.н., доцент Авсиевич А.В.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
1.1	Целью дисциплины «Системное программирование» является формирование компетенции ПК-1 и получение необходимых знаний, умений, навыков.
1.2	Задачами дисциплины «Системное программирование» является знакомство с методами и алгоритмами управления ресурсами, структурами данных и программными интерфейсами операционных систем современных персональных ЭВМ (ПЭВМ); практическое освоение программного интерфейса современных операционных систем, в частности, Win32 API, и современных инструментальных средств разработки приложений для ПЭВМ; изучение внутреннего интерфейса и недокументированных возможностей современных операционных систем; изучение управляющих структур данных операционных систем.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	Б1.В.08

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
ПК-1 Способен создавать инструментальные средства программирования	
ПК-1.1 Разрабатывает программный код на языках программирования низкого уровня	
ПК-1.2 Осуществляет отладку программ, написанных на языке низкого уровня	
06.028. Профессиональный стандарт "СИСТЕМНЫЙ ПРОГРАММИСТ", утверждённый приказом Министерства труда и социальной защиты Российской Федерации от 29 сентября 2020 г. N 678н (зарегистрирован Министерством юстиции Российской Федерации 26 ноября 2020 г., регистрационный N 60582)	
ПК-1. А. Разработка компонентов системных программных продуктов	
А/04.6 Создание инструментальных средств программирования	

В результате освоения дисциплины (модуля) обучающийся должен

3.1	Знать:
3.1.1	основные методы и алгоритмы управления ресурсами вычислительных систем; современные инструментальные средства разработки приложений для ПЭВМ; архитектуру и особенности работы современных микропроцессоров в реальном и защищенном режимах; особенности работы с памятью; особенности файловых систем.
3.1.2	способы разработки кроссплатформенных приложений; методы портирования приложений с одной платформы на другую; базовые принципы и современные методы алгоритмизации, написания программ и автономной отладки при программировании последовательных, параллельных, распределенных приложений, приложений реального времени; современные языки и средства.
3.2	Уметь:
3.2.1	использовать методы и алгоритмы управления ресурсами вычислительных систем; использовать современные инструментальные средства разработки приложений для ПЭВМ; использовать языки и системы программирования для решения профессиональных задач.
3.2.2	отлаживать и тестировать системные программы; использовать способы кроссплатформенной разработки приложений.
3.3	Владеть:
3.3.1	Владеть: навыками использования методов и алгоритмов управления ресурсами вычислительных систем; навыками использования современных инструментальных средства разработки приложений для ПЭВМ.
3.3.2	Владеть: навыками создания кроссплатформенных приложений; инструментальными средствами создания программных библиотек.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Примечание
	Раздел 1. Управление задачами в операционных системах.			
1.1	Назначение и функции операционной системы. Классификация операционных систем. /Лек/	5	1	
1.2	Структура многозадачной операционной системы. /Лек/	5	1	
1.3	Использование системных вызовов и API функций. /Лаб/	5	4	Практическая подготовка
1.4	Мониторинг состояния системы /Лаб/	5	4	Практическая подготовка
1.5	Процессы. /Лек/	5	2	
1.6	Работа с процессами. /Лаб/	5	4	Практическая подготовка

1.7	Межпроцессное взаимодействие. /Лек/	5	2	
1.8	Организация межпроцессного взаимодействия. /Лаб/	5	6	Практическая подготовка
1.9	Потоки. /Лек/	5	2	
1.10	Работа с потоками. /Лаб/	5	6	Практическая подготовка
1.11	Синхронизация. Примитивы синхронизации. /Лек/	5	2	
1.12	Синхронизация потоков. /Лаб/	5	4	Практическая подготовка
Раздел 2. Управление ресурсами вычислительной системы				
2.1	Файловая система. /Лек/	5	2	
2.2	Реализация сервисов операционной системы. /Лаб/	5	4	Практическая подготовка
2.3	Управление памятью. /Лек/	5	2	
2.4	Статические и динамические библиотеки. /Лек/	5	2	
2.5	Создание и использование статических библиотек. /Ср/	5	6	
2.6	Создание и использование динамических библиотек. /Ср/	5	5	
Раздел 3. Самостоятельная работа				
3.1	Подготовка к лекциям /Ср/	5	8	
3.2	Подготовка к лабораторным работам /Ср/	5	32	
Раздел 4. Контактные часы на аттестацию				
4.1	Зачет /КЭ/	5	0,15	

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Оценочные материалы для проведения промежуточной аттестации обучающихся приведены в приложении к рабочей программе дисциплины.

Формы и виды текущего контроля по дисциплине (модулю), виды заданий, критерии их оценивания, распределение баллов по видам текущего контроля разрабатываются преподавателем дисциплины с учетом ее специфики и доводятся до сведения обучающихся на первом учебном занятии.

Текущий контроль успеваемости осуществляется преподавателем дисциплины (модуля) в рамках контактной работы и самостоятельной работы обучающихся. Для фиксирования результатов текущего контроля может использоваться ЭИОС.

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

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л1.1	Власенко А. Ю., Карабцев С. Н., Рейн Т. С.	Операционные системы: учебное пособие	Кемерово : КеМГУ, 2019	http://e.lanbook.com/book/12
Л1.2	Малявко А. А.	Формальные языки и компиляторы: учебное пособие для вузов	Москва: Юрайт, 2020	https://urait.ru/bcode/45325

6.1.2. Дополнительная литература				
	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л2.1	Соколова В. В.	Вычислительная техника и информационные технологии. Разработка мобильных приложений: Учебное пособие для вузов	Москва: Юрайт, 2020	tps://urait.ru/bcode/45136
6.2 Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине (модулю)				
6.2.1 Перечень лицензионного и свободно распространяемого программного обеспечения				
6.2.1.1	Microsoft Windows10 Pro Договор №034210000481700004			
6.2.1.2	Microsoft office 2013 (Лицензия № 61887848) Договор на поставку № 0342100004813000011			
6.2.1.3	7-zip (http://www.7-zip.org/ (GNU LGPL license)Microsoft Windows10 Pro Договор №034210000481700004			
6.2.1.4	Microsoft office 2013 (Лицензия № 61887848) Договор на поставку № 03421			
6.2.2 Перечень профессиональных баз данных и информационных справочных систем				
6.2.2.1	Крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки- https://github.com/			
6.2.2.2	База книг и публикаций Электронной библиотеки "Наука и Техника" - http://www.n-t.ru			
6.2.2.3	Портал для разработчиков электронной техники: http://www.espec.ws/			
6.2.2.4	База данных «Библиотека программиста» https://proglib.io/			
6.2.2.5	База данных «Отраслевой портал специалистов» http://www.connect-wit.ru/			
6.2.2.6	Гарант.ру https://www.garant.ru/			
6.2.2.7	КонсультантПлюс http://www.consultant.ru/			
7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
7.1	мультимедийное оборудование для предоставления учебной информации большой аудитории и/или звукоусиливающее оборудование (стационарное или переносное).			
7.2	Учебные аудитории для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, укомплектованные специализированной мебелью и техническими средствами обучения: мультимедийное оборудование и/или звукоусиливающее оборудование (стационарное или переносное)			
7.3	Помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета.			
7.4	Помещения для хранения и профилактического обслуживания учебного оборудования			

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Системное программирование

(наименование дисциплины(модуля))

Направление подготовки / специальность

09.03.02 Информационные системы и технологии

(код и наименование)

Направленность (профиль)/специализация

Информационные системы и технологии на транспорте

(наименование)

Форма обучения

Очное

Семестр 5 (зачет)

Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Формы промежуточной аттестации: **зачет в 5 семестре.**

Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции
ПК-1 Способен создавать инструментальные средства программирования	ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня
	ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня

Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся знает: основные методы и алгоритмы управления ресурсами вычислительных систем; современные инструментальные средства разработки приложений для ПЭВМ; архитектуру и особенности работы современных микропроцессоров в реальном и защищенном режимах; особенности работы с памятью; особенности файловых систем.	Вопросы (1-10)
	Обучающийся умеет: использовать методы и алгоритмы управления ресурсами вычислительных систем; использовать современные инструментальные средства разработки приложений для ПЭВМ; использовать языки и системы программирования для решения профессиональных задач	Вопросы (8-19)
	Обучающийся владеет: навыками использования методов и алгоритмов управления ресурсами вычислительных систем; навыками использования современных инструментальных средства разработки приложений для ПЭВМ	Вопросы (14-25)
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся знает: способы разработки кроссплатформенных приложений; методы портирования приложений с одной платформы на другую; базовые принципы и современные методы алгоритмизации, написания программ и автономной отладки при программировании последовательных, параллельных, распределенных приложений, приложений реального времени; современные языки и средства	Вопросы (20-27)
	Обучающийся умеет: отлаживать и тестировать системные программы; использовать способы кроссплатформенной разработки приложений.	Вопросы (25-38)
	Обучающийся владеет: навыками создания кроссплатформенных приложений; инструментальными средствами создания программных библиотек.	Вопросы (38-48)

Промежуточная аттестация (зачет) проводится в одной из следующих форм:

- 1) ответ на билет, состоящий из теоретических вопросов и практических заданий;
- 2) выполнение заданий в ЭИОС Университета.

2. Типовые¹ контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

2.1 Типовые вопросы (тестовые задания) для оценки знаниевого образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся знает: основные методы и алгоритмы управления ресурсами вычислительных систем; современные инструментальные средства разработки приложений для ПЭВМ; архитектуру и особенности работы современных микропроцессоров в реальном и защищенном режимах; особенности работы с памятью; особенности файловых систем.
<i>Примеры вопросов/заданий</i> 1. Что относится к ресурсам ЭВМ. 2. Организация памяти ЭВМ. 3. Оперативная память как ресурс. 4. Система регистров микропроцессора и их использование в качестве ресурсов. 5. Система адресации в современных компьютерах. 6. Процессы. 7. Потоки. 8. Алгоритмы работы с памятью. 9. Алгоритмы планирования потоков. 10. Диспетчеризация. 11. Многопоточные приложения.	
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся знает: способы разработки кроссплатформенных приложений; методы портирования приложений с одной платформы на другую; базовые принципы и современные методы алгоритмизации, написания программ и автономной отладки при программировании последовательных, параллельных, распределенных приложений, приложений реального времени; современные языки и средства
<i>Примеры вопросов/заданий</i> 1. Понятие о библиотеках функций. 2. Статические библиотеки. 3. Динамические библиотеки. 4. Способы компиляции, сборки и использования статических библиотек. 5. Способы компиляции, сборки и использования динамических библиотек. 6. Сборка многофайловых программ. 7. Основы кроссплатформенной разработки приложений. 8. Переносимость приложений на уровне исходных кодов. 9. Использование для разработки приложений кроссплатформенных библиотек. 10. Способы условной компиляции. Использование препроцессора 11. Стандартная библиотека функций. 12. Использование стандартной библиотеки для программирования многопоточных приложений.	

¹ Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств.

2.2 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся умеет: использовать методы и алгоритмы управления ресурсами вычислительных систем; использовать современные инструментальные средства разработки приложений для ПЭВМ; использовать языки и системы программирования для решения профессиональных задач
<i>Примеры заданий</i> 1. Системная программа – это: a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования. c. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации. d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память. 2. Прикладная программа – это a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации. b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования. c. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации. d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память. 3. Программный модуль — это a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации. b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования. c. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации. d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память. 4. Объектный модуль — это a. программный модуль, представленный в форме, пригодной для загрузки в оперативную память для выполнения. b. программный модуль, получаемый в результате трансляции исходного модуля. c. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память. d. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции. 5. Программа обслуживания (утилиты) — это a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации. b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования.	

с. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации. d. программа или функционально завершённый фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.	
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся владеет: навыками использования методов и алгоритмов управления ресурсами вычислительных систем; навыками использования современных инструментальных средства разработки приложений для ПЭВМ
1. Методы и алгоритмы управления ресурсами вычислительной системы. 2. Использование многопоточности в приложениях. 3. Управление процессами. 4. Управление потоками. 5. Компиляция многопоточных приложений. 6. Использование командной строки для создания многопоточных приложений. Задание 1. Выполнить проектирование многопоточного приложения используя нотацию UML. 2. Произвести компиляцию приложения, используя командную строку. 3. При необходимости исправить ошибки и повторить компиляцию. 4. Выполнить запуск программы и убедиться в ее работоспособности.	
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся умеет: отлаживать и тестировать системные программы; использовать способы кроссплатформенной разработки приложений.
1. Особенности работы с памятью программ. Операторы выделения блоков памяти и удаления выделенных блоков памяти. 2. Особенности работы с процессами. Создание процесса. Удаление процесса. 3. Особенности работы с потоками. Создание потоков и управление ими. 4. Использование средств управления потоками. 5. Семафоры и мьютексы. 6. Использование API функций операционной системы Windows. 7. Использование системных вызовов операционной системы Linux. 8. Особенности тестирования системных программ. 9. Особенности компиляции программ на уровне ядра операционной системы.	
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся владеет: навыками создания кроссплатформенных приложений; инструментальными средствами создания программных библиотек.
1. Использование стандартной библиотеки для создания кроссплатформенных многопоточных приложений. 2. Ключевые режимы компилятора для создания системных приложений. 3. Особенности программирования программных библиотек в операционной среде Windows. 4. Особенности программирования библиотек в операционной системе Linux. 5. Особенности создания сервисных программ в операционных системах Windows и Linux. 6. Вызов библиотечных функций и их использование в программах. 7. Компиляция программы с включением в нее отладочной информации. 8. Межпроцессное взаимодействие программ. 9. Взаимодействие программных потоков.	

2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации

1. Назначение, состав и функции ОС.
2. Вычислительные процессы и ресурсы. Виды ресурсов.
3. Диаграмма состояний вычислительного процесса в ОС общего назначения и в ОС реального времени.
4. Понятие процесса Win32. Операции, выполняемые с процессами.
5. Мониторинг процессов и потоков Win32
6. Диспетчеризация потоков в Win32.
7. Потоки в Win32. Операции выполняемые с потоками.
8. Использование потоков при разработке приложений.
9. Квантование времени в MS Windows. Определение величины кванта времени.
10. Независимые и взаимодействующие потоки.
11. Понятие критического участка и критической секции.
12. Средства синхронизации потоков Win32.
13. Задача поставщик-потребитель и ее решение с помощью операций P(S) и V(S).
14. Объекты ядра MS Windows, их создание и использование.
15. Мьютексы, как средство синхронизации потоков.
16. Семафоры и их использование.
17. События и их использование.
18. Функции ожидания одного и нескольких событий
19. Сегментная и страничная адресация памяти. Сравнительный анализ
20. Deskriptorные таблицы. Схема вычисления физического адреса в режиме 286.
21. Формат дескрипторов в защищенных режимах 286 и 386.
22. Байт доступа. Поле доступа. Назначение битов.
23. Условие доступа программы к сегменту данных.
24. Реализация защиты ОС от прикладных программ.
25. Кольца защиты. Шлюзы вызова и задачи.
26. Аппаратная поддержка многозадачности. Сегмент состояния задачи.
27. Обработка прерываний в защищенном режиме.
28. Модель Холта.
29. Понятие тупика. Пример тупика процессов, использующих семафоры.
30. Необходимые условия возникновения тупика.
31. Методы борьбы с тупиками.
32. Управление страничной памятью. Стратегии подкачки и рабочие наборы страниц.
33. Разделы в виртуальном адресном пространстве процесса. Адресное пространство процесса MS Windows.
34. Управление памятью с помощью AWE.
35. Получение информации о состоянии виртуальной памяти
36. Резервирование регионов в адресном пространстве и передача физической памяти региону. Освобождение регионов
37. Кучи. Необходимость создания дополнительных куч. Операции с кучами.
38. Файлы, проецируемые в память
39. Динамически загружаемые библиотеки. Области применения.
40. Основные DLL MS Windows. Достоинства и недостатки DLL.
41. Использование DLL - импорт функций из DLL.
42. Явная и неявная загрузка DLL.
43. Эволюция архитектуры файловых систем.
44. Файловые системы Win32.
45. Структура магнитного диска.
46. Файловая система FAT.
47. Файловая система NTFS.
48. Основные отличия FAT и NTFS.

3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90 % от общего объёма заданных вопросов;
- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76 % от общего объёма заданных вопросов;
- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы – 75–60 % от общего объёма заданных вопросов;
- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60 % от общего объёма заданных вопросов.

Критерии формирования оценок по результатам выполнения заданий

«Отлично/зачтено» – ставится за работу, выполненную полностью без ошибок и недочетов.

«Хорошо/зачтено» – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

«Удовлетворительно/зачтено» – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

«Неудовлетворительно/не зачтено» – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

Виды ошибок:

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*
- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*
- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

Критерии формирования оценок по результатам выполнения лабораторных работ

«Зачтено» – ставится за работу, выполненную полностью без ошибок и недочетов в соответствии с заданием, выданным для выполнения лабораторной работы. Обучающийся полностью владеет информацией по теме работы, решил все поставленные в задании задачи.

«Не зачтено» - ставится за работу, если обучающийся правильно выполнил менее 2/3 всей работы, использовал при выполнении работы неправильные алгоритмы, допустил грубые ошибки при расчетах, сформулировал неверные выводы по результатам работы.

Критерии формирования оценок по результатам выполнения практических работ

«Зачтено» – ставится за работу, выполненную полностью без ошибок и недочетов в соответствии с заданием. Обучающийся полностью владеет информацией по теме работы, решил все поставленные в задании задачи.

«Не зачтено» - ставится за работу, если обучающийся правильно выполнил менее 2/3 всей работы, использовал при выполнении работы неправильные алгоритмы, допустил грубые ошибки при расчетах, сформулировал неверные выводы по результатам работы.

Критерии формирования оценок по зачету

«Зачтено» - обучающийся демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки, освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при ответе, достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности.

«Не зачтено» - выставляется в том случае, когда обучающийся демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. У экзаменуемого слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки, допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки.