МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Луганский государственный университет имени Владимира Даля»
(ФГБОУ ВО «ЛГУ им. В. Даля»)

Краснодонский факультет инженерии и менеджмента (филиал)
Кафедра информационных технологий и транспорта

УТВЕРЖДАЮ:
Директор
_____ Панайотов К.К.
(подпись)
«21» апреля 2023 года

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

По дисциплине Системное программирование
(название дисциплины по учебному плану)

По направлению подготовки 09.03.01 – Информатика и вычислительная техника
(код, название без кавычек)

Профиль подготовки Компьютерные системы и сети

Краснодон 2023

Подписано цифровой подписью: Панайотов Константин Константинович
Причина: Я являюсь автором этого документа
Дата: 10 Март 2025 г. 9:11:20
DN: E=uc_fk@roskazna.ru, S=77 Москва, ИНН ЮЛ=7710568760, ОГРН=1047797019830, STREET="Большой Златоустинский переулок, д. 6, строение 1", L=г. Москва, C=RU, O=Казначейство России, CN=Федеральное казначейство

Лист согласования РПУД

Рабочая программа учебной дисциплины «Системное программирование» по направлению подготовки 09.03.01 Информатика и вычислительная техника, профиль «Компьютерные системы и сети» – 24 с.

Рабочая программа учебной дисциплины «Системное программирование» разработана в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки Российской Федерации от «19» сентября 2017 года № 929.

СОСТАВИТЕЛЬ (СОСТАВИТЕЛИ):
к.т.н., доц. Панайотов К.К.

(ученая степень, ученое звание, должность фамилия, инициалы)

Рабочая программа дисциплины утверждена на заседании кафедры информационных технологий и транспорта «15» марта 2023 г., протокол № 7.

Заведующий кафедрой _____ Бихдрикер А.С.

Рекомендована на заседании учебно-методической комиссии факультета «20» марта 2023 г., протокол № 8.

Председатель учебно-методической комиссии факультета _____ Замота О.Н.

© Панайотов К.К., 2023 год
© ФГБОУ ВО «ЛГУ им. В. ДАЛЯ» КФИиМ (филиал), 2023 год

Структура и содержание дисциплины

1. Цели и задачи учебной дисциплины, ее место в учебном процессе

Цель изучения дисциплины – ознакомиться с теоретическими основами, методами, средствами разработки и программирования ОС и ее низкоуровневых структурных элементов.

Задачи: изучение организации и принципов построения современных операционных систем и системных программ; формирование представлений об общей методологии разработки системно-ориентированных программ с использованием современных алгоритмических языков и систем программирования; изучение основ программирования прикладных интерфейсов.

2. Место дисциплины в структуре ООП

Дисциплина входит в вариативную часть профессионального цикла дисциплин подготовки студентов по направлению 09.03.01 Информатика и вычислительная техника

Основывается на базе дисциплин: программирование; архитектура компьютера.

Является основой для изучения следующих дисциплин: системное программное обеспечение; ЭВМ и периферийные устройства.

3. Результаты освоения дисциплины

Код и наименование компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Перечень планируемых результатов
ПК-2 Способен проектировать и разрабатывать компоненты программных комплексов информационных систем	ПК-2.2 Способен разрабатывать и интегрировать компоненты программных комплексов информационных систем	Знает: современные методы разработки и реализации математических моделей Умеет: применять современные методы разработки и реализации математических моделей Владеет: практическими навыками применения современных методов разработки и реализации математических моделей

4. Структура и содержание дисциплины

4.1. Структура учебной дисциплины

Вид учебной работы	**Объем часов**		
	Очная форма	**Очно-заочная форма**	**Заочная форма**
Общая учебная нагрузка (всего)	**180**		**180**
Обязательная аудиторная учебная нагрузка (всего)	**68**		**12**
в том числе:			
Лекции	34		6
Семинарские занятия	-		-
Практические занятия	-		-
Лабораторные работы	34		6
Курсовая работа (курсовой проект)	-		-
Другие формы и методы организации образовательного процесса (*расчетно-графические работы, групповые дискуссии, ролевые игры, тренинг, компьютерные симуляции, интерактивные лекции, семинары, анализ деловых ситуаций и т.п.)*	-		-
Самостоятельная работа студента (всего)	112		168
Итоговая аттестация	экзамен		экзамен

4.2. Содержание разделов дисциплины:

Тема 1. УПРАВЛЕНИЕ ПОТОКАМИ И ПРОЦЕССАМИ

Определение потока. Контекст потока. Состояния потока. Диспетчеризация и планирование потоков. Определение процесса. Определение потока. Создание потоков. Завершение потоков. Приостановка и возобновление потоков. Псевдодескрипторы потоков. Обработка ошибок в Windows. Определение процесса. Создание процессов. Завершение процессов. Наследование дескрипторов. Дублирование дескрипторов. Псевдодескрипторы процессов. Обслуживание потоков. Динамическое изменение приоритетов потоков.

Тема 2. СИНХРОНИЗАЦИЯ ПОТОКОВ И ПРОЦЕССОВ

Синхронизация. Непрерывные действия и команды. Определение синхронизации. Программная реализация синхронизации. Аппаратная реализация синхронизации. Примитивы синхронизации. Синхронизация потоков в Windows. Критические секции. Объекты синхронизации и функции ожидания. Мьютексы. События Семафоры Взаимоисключающий доступ к переменным. Атомарные операции. Замена значения переменной. Условная замена значения переменной. Инкремент и декремент переменной. Изменение значения переменной Тупики. Определение тупиков. Классификация

системных ресурсов. Обнаружение тупиков. Восстановление заблокированного процесса. Предотвращение тупиков. Безопасное завершение потоков в Windows.

Тема 3. ПРОГРАММИРОВАНИЕ КОНСОЛЬНЫХ ПРИЛОЖЕНИЙ

Структура консольного приложения. Структура консоли. Входной буфер консоли. Буфер экрана. Работа с консолью. Создание консоли. Освобождение консоли. Стандартные дескрипторы ввода-вывода. Работа с окном консоли. Получение дескриптора окна консоли. Получение и изменение заголовка консоли. Определение максимального размера окна. Установка координат окна. Работа с буфером экрана. Создание и активация буфера экрана. Определение и установка параметров буфера экрана. Функции для работы с курсором. Чтение и установка атрибутов консоли. Ввод-вывод на консоль. Ввод-вывод высокого уровня. Ввод низкого уровня. Вывод низкого уровня. Режимы ввода-вывода консоли. Прокрутка буфера экрана.

Тема 4. ОБМЕН ДАННЫМИ МЕЖДУ ПАРАЛЛЕЛЬНЫМИ ПРОЦЕССАМИ

Передача данных. Способы передачи данных между процессами. Связи между процессами. Передача сообщений. Синхронный и асинхронный обмен данными. Буферизация. Работа с анонимными каналами в Windows Анонимные каналы. Создание анонимных каналов. Соединение клиентов с анонимным каналом. Обмен данными по анонимному каналу. Примеры работы с анонимными каналами. Перенаправление стандартного ввода-вывода. Работа с именованными каналами в Windows. Именованные каналы. Создание именованных каналов. Соединение сервера с клиентом. Соединение клиентов с именованным каналом. Обмен данными по именованному каналу. Копирование данных из именованного канала. Передача транзакций по именованному каналу. Определение и изменение состояния именованного канала. Получение информации об именованном канале. Работа с почтовыми ящиками в Windows. Концепция почтовых ящиков Создание почтовых ящиков. Соединение клиентов с почтовым ящиком. Обмен данными через почтовый ящик. Получение информации о почтовом ящике. Изменение времени ожидания сообщения.

Тема 5. СТРУКТУРНАЯ ОБРАБОТКА ИСКЛЮЧЕНИЙ

Фреймовая обработка исключений. Исключения и их обработчики. Получение кода исключения. Функции фильтра. Получение информации об исключении. Генерация программных исключений. Необработанные исключения. Обработка исключений с плавающей точкой. Обработка вложенных исключений. Передача управления и выход из фрейма. Встраивание SEH в механизм исключений C++ . Финальная обработка исключений. Финальные блоки фрейма. Проверка завершения фрейма. Обработка вложенных финальных блоков/

Тема 6. РАБОТА С ВИРТУАЛЬНОЙ ПАМЯТЬЮ

Виртуальная память. Концепция виртуальной памяти. Организация виртуальной памяти. Алгоритмы замещения страниц. Рабочее множество процесса. Организация виртуальной памяти в Windows. Работа с виртуальной памятью в Windows. Состояния виртуальной памяти процесса. Резервирование, распределение и освобождение виртуальной памяти. Блокирование виртуальных страниц в реальной памяти. Изменение атрибутов доступа к виртуальной странице. Управление рабочим множеством страниц процесса. Инициализация и копирование блоков виртуальной памяти. Определение состояния памяти. Работа с виртуальной памятью в другом процессе. Работа с кучей в Windows. Создание и удаление кучи. Распределение и освобождение памяти из кучи. Перераспределение памяти из кучи. Блокирование и разблокирование кучи. Проверка состояния кучи. Уплотнение кучи.

Тема 7. УПРАВЛЕНИЕ ФАЙЛАМИ

Общие концепции. Накопители на жестких магнитных дисках. Секторы и кластеры. Форматирование дисков. Функции файловой системы. Каталоги. Буферизация ввода-вывода. Кэширование ввода-вывода. Работа с файлами в Windows. Именование файлов в Windows. Создание и открытие файлов. Закрытие и удаление файлов. Запись данных в файл. Освобождение буферов файла. Чтение данных из файла. Копирование файла. Перемещение файла. Замещение файла. Работа с указателем позиции файла. Определение и изменение атрибутов файла. Определение и изменение размеров файла. Блокирование файла. Получение информации о файле. Работа с каталогами (папками) в Windows. Создание каталога. Поиск файлов в каталоге. Удаление каталога. Перемещение каталога. Определение и установка текущего каталога. Наблюдение за изменениями в каталоге.

Тема 8. АСИНХРОННАЯ ОБРАБОТКА ДАННЫХ

Асинхронный вызов процедур. Механизм асинхронного вызова процедур. Установка асинхронных процедур. Приостановка потока. Ожидание события. Оповещение и ожидание события. Асинхронный доступ к данным. Концепция асинхронного ввода-вывода. Асинхронная запись данных. Асинхронное чтение данных. Блокирование файлов. Определение состояния асинхронной операции ввода-вывода. Отмена асинхронной операции ввода-вывода. Процедуры завершения ввода-вывода. Асинхронная запись данных с процедурами завершения. Асинхронное чтение данных с процедурами завершения. Порты завершения. Концепция порта завершения. Создание порта завершения. Получение пакета из порта завершения. Посылка пакета в порт завершения. Работа с ожидающим таймером. Ожидающий таймер Создание ожидающего таймера. Установка ожидающего таймера. Отмена ожидающего таймера. Открытие существующего ожидающего таймера. Процедуры завершения ожидания.

Тема 9. ДИНАМИЧЕСКИ ПОДКЛЮЧАЕМЫЕ БИБЛИОТЕКИ

Отображение файлов в память. Концепция механизма отображения файлов в память. Создание и открытие объекта, отображающего файл. Отображение файла в память. Обмен данными между процессами через отображаемый в память файл. Сброс вида в файл. Динамически подключаемые библиотеки. Концепция динамически подключаемых библиотек. Создание DLL. Динамическая загрузка и отключение DLL. Использование DLL. Использование файла определений. Статическая загрузка DLL. Локальная память потока. Динамическая локальная память потока. Распределение и освобождение локальной памяти потока. Запись и чтение из локальной памяти потока. Статическая локальная память потока

Тема 10. РАЗРАБОТКА СЕРВИСОВ В WINDOWS

Сервисы в Windows. Концепция сервиса. Структура сервиса. Организация функции main. Организация функции ServiceMain. Организация обработчика управляющих команд. Работа с сервисами в Windows. Открытие доступа к базе данных сервисов. Установка сервиса. Открытие доступа к сервису. Запуск сервиса. Определение и изменение состояния сервиса. Определение и изменение конфигурации сервиса. Определение имени сервиса. Управление сервисом. Удаление сервисов. Блокирование базы данных сервисов.

4.3. Лекции

№ п/п	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Тема 1. Управление потоками и процессами	3	2
2	Тема 2. Синхронизация потоков и процессов	3	2
3	Тема 3. Анализ требований и определение спецификаций программного обеспечения	3	2
4	Тема 4. Обмен данными между параллельными процессами	3	
5	Тема 5. Структурная обработка исключений	3	
6	Тема 6. Работа с виртуальной памятью	3	
7	Тема 7. Управление файлами	4	
8	Тема 8. Асинхронная обработка данных	4	
9	Тема 9. Динамически подключаемые библиотеки	4	
10	Тема 10. Разработка сервисов в windows	4	
Итого:		34	6

4.4. Практические занятия

Планом не предусмотрены.

4.5. Лабораторные работы

№ п/п	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Лабораторная работа № 1 Работа с компилятором GCC	1	2
2	Лабораторная работа № 2 Работа с утилитой MAKE	1	2
3	Лабораторная работа № 3 Работа с указателями в языке программирования C	2	2
5	Лабораторная работа № 4 Работа со статическими библиотеками в языке программирования C	2	
6	Лабораторная работа № 5 Работа с динамическими библиотеками в языке программирования C	2	
7	Лабораторная работа № 6 Изучение модели COM на примере простейшего приложения	2	
9	Лабораторная работа № 7 Разработка двухкомпонентного COM/DLL-сервера	2	
10	Лабораторная работа № 8 Структурная обработка исключений	2	
11	Лабораторная работа № 9 Управление пользователями и группами	2	
13	Лабораторная работа № 10 Работа с файлами в языке программирования C	3	
14	Лабораторная работа № 11 Повторная применимость COM-компонентов: включение	3	
15	Лабораторная работа № 12 Повторная применимость COM-компонентов: агрегирование	3	
16	Лабораторная работа № 13 Асинхронный ввод-вывод	3	
17	Лабораторная работа № 14 Применение портов завершения	3	
18	Лабораторная работа № 15 Разработка Windows-сервиса	3	
Итого:		34	6

4.6. Самостоятельная работа студентов

№ п/п	Название темы	Вид СРС	Объем часов	
			Очная форма	Заочная форма
1	Тема 1. Управление потоками и процессами	подготовка к лабораторным работам и оформление отчетов,	10	16
2	Тема 2. Синхронизация потоков и процессов	подготовка к лабораторным работам и оформление отчетов,	10	16
3	Тема 3. Анализ требований и определение спецификаций программного обеспечения	подготовка к лабораторным работам и оформление отчетов,	10	16
4	Тема 4. Обмен данными между параллельными процессами	подготовка к лабораторным работам и оформление отчетов,	10	16
5	Тема 5. Структурная обработка исключений	подготовка к лабораторным работам и оформление отчетов,	10	14
6	Тема 6. Работа с виртуальной памятью	подготовка к лабораторным работам и оформление отчетов,	14	18
	Тема 7. Управление файлами	подготовка к лабораторным работам и оформление отчетов,	12	18
	Тема 8. Асинхронная обработка данных	подготовка к лабораторным работам и оформление отчетов,	12	18
	Тема 9. Динамически подключаемые библиотеки	подготовка к лабораторным работам и оформление отчетов,	12	18
	Тема 10. Разработка сервисов в windows	подготовка к лабораторным работам и оформление отчетов,	12	18
Итого:			112	168

4.7. Курсовые работы/проекты.

Планом не предусмотрены

5. Образовательные технологии

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

- традиционные объяснительно-иллюстративные технологии, которые обеспечивают доступность учебного материала для большинства студентов, системность, отработанность организационных форм и привычных методов, относительно малые затраты времени;
- технологии проблемного обучения, направленные на развитие познавательной активности, творческой самостоятельности студентов и предполагающие последовательное

и целенаправленное выдвижение перед студентом познавательных задач, разрешение которых позволяет студентам активно усваивать знания (используются поисковые методы; постановка познавательных задач);

- технологии развивающего обучения, позволяющие ориентировать учебный процесс на потенциальные возможности студентов, их реализацию и развитие;
- технологии концентрированного обучения, суть которых состоит в создании максимально близкой к естественным психологическим особенностям человеческого восприятия структуры учебного процесса и которые дают возможность глубокого и системного изучения содержания учебных дисциплин за счет объединения занятий в тематические блоки;
- технологии модульного обучения, дающие возможность обеспечения гибкости процесса обучения, адаптации его к индивидуальным потребностям и особенностям обучающихся (применяются, как правило, при самостоятельном обучении студентов по индивидуальному учебному плану);
- технологии дифференцированного обучения, обеспечивающие возможность создания оптимальных условий для развития интересов и способностей студентов, в том числе и студентов с особыми образовательными потребностями, что позволяет реализовать в культурно-образовательном пространстве университета идею создания равных возможностей для получения образования
- технологии активного (контекстного) обучения, с помощью которых осуществляется моделирование предметного, проблемного и социального содержания будущей профессиональной деятельности студентов (используются активные и интерактивные методы обучения) и т.д.

Максимальная эффективность педагогического процесса достигается путем конструирования оптимального комплекса педагогических технологий и (или) их элементов на личностно-ориентированной, деятельностной, диалогической основе и использования необходимых современных средств обучения.

6. Учебно-методическое и информационное обеспечение дисциплины

Основная литература:

1. Зайцев М.Г., Объектно-ориентированный анализ и программирование : учебное пособие / Зайцев М.Г. - Новосибирск : Изд-во НГТУ, 2017. - 84 с. - ISBN 978-5-7782-3308-9 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN9785778233089.html

2. Иванова Г.С., Технология программирования / Иванова Г.С. - М. : Издательство МГТУ им. Н. Э. Баумана, . - 336 с. (Сер. Информатика в техническом университете) - ISBN 5-7038-2891-0 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN5703828910.html

3. Подбельский В.В., Язык C#. Базовый курс : учеб. пособие / В.В. Подбельский. - 2-е изд., перераб. и доп. - М. : Финансы и статистика, 2015. - 408 с. - ISBN 978-5-279-03534-2 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN9785279035342.html

4. Самохвалов Э.Н., Введение в проектирование и разработку приложений на языке программирования C# : учебное пособие / Э.Н. Самохвалов, Г.И. Ревунков, Ю.Е. Гапанюк - М. : Издательство МГТУ им. Н. Э. Баумана, 2018. - 244 с. - ISBN 978-5-7038-4553-0 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN9785703845530.html

Дополнительная литература:

1. Комлев Н.Ю., Объектно Ориентированное Программирование. Настольная книга программиста / Комлев Н. Ю. - М. : СОЛОН-ПРЕСС, 2018. - 298 с. - ISBN 978-5-91359-276-7 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL :

http://www.studentlibrary.ru/book/ISBN9785913592767.html

2. Лисицин Д.В., Объектно-ориентированное программирование / Лисицин Д.В. - Новосибирск : Изд-во НГТУ, 2010. - 88 с. - ISBN 978-5-7782-1454-5 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN9785778214545.html

3. Суханов М.В., Основы Microsoft .NET Framework и языка программирования C#: / Суханов М.В. - Архангельск : ИД САФУ, 2014. - 96 с. - ISBN 978-5-261-00934-4 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN9785261009344.html

4. Уйманова Н.А., Основы объектно-ориентированного программирования / Уйманова Н.А. - Оренбург: ОГУ, 2017. - 768 с. - ISBN 978-5-7410-1993-1 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN9785741019931.html

в) интернет-ресурсы:

Министерство образования и науки Российской Федерации – http://минобрнауки.рф/

Федеральная служба по надзору в сфере образования и науки – http://obrnadzor.gov.ru/

Министерство образования и науки Луганской Народной Республики – https://minobr.su

Народный совет Луганской Народной Республики – https://nslnr.su

Портал Федеральных государственных образовательных стандартов высшего образования – http://fgosvo.ru

Федеральный портал «Российское образование» – http://www.edu.ru/

Информационная система «Единое окно доступа к образовательным ресурсам» – http://window.edu.ru/

Федеральный центр информационно-образовательных ресурсов – http://fcior.edu.ru/

Электронные библиотечные системы и ресурсы

Электронно-библиотечная система «Консультант студента» – http://www.studentlibrary.ru/cgi-bin/mb4x

Электронно-библиотечная система «StudMed.ru» –https://www.studmed.ru

Информационный ресурс библиотеки образовательной организации

Научная библиотека имени А. Н. Коняева – http://biblio.dahluniver.ru/

7. Материально-техническое обеспечение дисциплины

Освоение дисциплины «Программирование» предполагает использование академических аудиторий, соответствующих действующим санитарным и противопожарным правилам и нормам.

Прочее:

- рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;
- для проведения лекционных занятий требуется аудитория на курс, оборудованная мультимедийным проектором с экраном;
- для проведения лабораторных работ требуется компьютерный класс, подключенный к Интернет.

Программное обеспечение:

Функциональное назначение	**Бесплатное программное обеспечение**	**Ссылки**
Офисный пакет	Libre Office 6.3.1	https://www.libreoffice.org/ https://ru.wikipedia.org/wiki/LibreOffice
Операционная система	UBUNTU 19.04	https://ubuntu.com/ https://ru.wikipedia.org/wiki/Ubuntu
Браузер	Firefox Mozilla	http://www.mozilla.org/ru/firefox/fx
Браузер	Opera	http://www.opera.com
Почтовый клиент	Mozilla Thunderbird	http://www.mozilla.org/ru/thunderbird
Файл-менеджер	Far Manager	http://www.farmanager.com/download.php
Архиватор	7Zip	http://www.7-zip.org/
Графический редактор	GIMP (GNU Image Manipulation Program)	http://www.gimp.org/ http://gimp.ru/viewpage.php?page_id=8 http://ru.wikipedia.org/wiki/GIMP
Редактор PDF	PDFCreator	http://www.pdfforge.org/pdfcreator
Аудиоплейер	VLC	http://www.videolan.org/vlc/

8. Оценочные средства по учебной дисциплине

Паспорт фонда оценочных средств по учебной дисциплине

Системное программирование

(наименование учебной дисциплины)

Перечень компетенций (элементов компетенций), формируемых в результате освоения учебной дисциплины (модуля) или практики

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Контролируемые темы учебной дисциплины, практики	Этапы формирования (семестр изучения)
1	ПК-2	Способен проектировать и разрабатывать компоненты программных комплексов информационных систем	ПК-2.2 Способен разрабатывать и интегрировать компоненты программных комплексов информационных систем	Тема 1. Управление потоками и процессами	3
				Тема 2. Синхронизация потоков и процессов	3
				Тема 3. Анализ требований и определение спецификаций программного обеспечения	3
				Тема 4. Обмен данными между параллельными процессами	3
				Тема 5. Структурная обработка исключений	3
				Тема 6. Работа с виртуальной памятью	3
				Тема 7. Управление файлами	3
				Тема 8. Асинхронная обработка данных	3
				Тема 9. Динамически подключаемые библиотеки	3
				Тема 10. Разработка сервисов в windows	3

Показатели и критерии оценивания компетенций, описание шкал оценивания

№ п/п	Код контролируемой компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Перечень планируемых результатов	Контролируемые темы учебной дисциплины	Наименование оценочного средства
1	ПК-2	ПК-2.2	Знает: современные методы разработки и реализации математических моделей Умеет: применять современные методы разработки и реализации математических моделей Владеет: практическими навыками применения современных методов разработки и реализации математических моделей	Тема 1. Тема 2. Тема 3. Тема 4. Тема 5. Тема 6. Тема 7 Тема 8. Тема 9. Тема 10.	Устный опрос, контрольная работа (по вариантам), тесты

Перечень вопросов (для проведения собеседования (устный или письменный опрос))

1. Основные принципы технологии .NET.
2. Что представляет собой платформа Visual Studio.NET?
3. Как создать консольное приложение?
4. Принципы объектно-ориентированного программирования.
5. Что представляют собой методы?
6. Как объявляется метод?
7. Какова область действия параметров метода?
8. Как вызываются методы?
9. Общие (статические) методы класса.
10. Как описываются классы в C#?
11. Что относится к членам класса?
12. Что такое статические члены класса?
13. Данные: поля и константы.
14. Спецификаторы полей и констант класса.
15. Как передаются параметры в методы?
16. Для чего предназначен параметр params?
17. Что представляет собой конструктор? Для чего он используется?
18. Какие бывают конструкторы?
19. Может ли класс не иметь конструктора?
20. Для чего предназначена система сбора мусора?
21. Что понимается под массивом?
22. Каковы возможные способы описания массивов (одномерных и многомерных)?
23. В каких случаях целесообразно описывать двумерный массив с помощью одномерных?
24. Какие типы допустимы для описания индексов массивов?
25. Какие типы могут использоваться в качестве базовых для описания массивов?

26. Как осуществляется ввод и вывод массивов?
27. Для чего предназначен цикл foreach?
28. Можно ли использовать цикл foreach для ввода элементов массива?
29. Как определяется базовый тип индексатора?
30. Что записывается в качестве имени индексатора?
31. Что содержит список параметров индексатора?
32. Что представляет собой перегрузка методов?
33. Что представляет собой перегрузка операций
34. Формат описания операции класса.
35. Какие операции нельзя перегружать?
36. Что является результатом перегрузки унарных операций?
37. Какие параметры могут быть у бинарных операций класса?
38. Как выполняется перегрузка операций отношения?
39. Чем являются строки в C#?
40. Какие операции определены для строк?
41. Как создаются строки?
42. Можно ли изменять значение строки?
43. В чем состоит принцип наследования?
44. Какие члены класса наследуются?
45. Что представляет собой защищенный доступ?
46. Как происходит вызов конструкторов базового класса?
47. Что такое сокрытие имен при наследовании?
48. Как получить доступ к сокрытому члену базового класса?
49. Что означает принцип полиморфизма?
50. Для чего используется позднее связывание?
51. В каких случаях используются виртуальные методы?
52. Какие условия необходимо соблюдать при переопределении виртуального метода?
53. Что представляют собой абстрактные классы? Для чего они предназначены?
54. Могут ли в абстрактном классе быть неабстрактные методы?
55. Как описывается интерфейс? Его назначение.
56. Какие члены может содержать интерфейс?
57. Какие спецификаторы допустимы у методов, реализующих интерфейс?
58. В каких случаях используется явная реализация интерфейса?
59. Как осуществляется наследование интерфейсов?
60. Можно ли явно реализованные методы объявлять виртуальными?
61. Можно ли повторно реализовать интерфейс, указав его имя в списке предков класса наряду с классом-предком?
62. Какие стандартные интерфейсы используются для работы с коллекциями?
63. Чем отличаются интерфейсы IComparable и IComparer?
64. Перечислите основные характеристики программ.
65. Приведите существующую классификацию программного обеспечения.
66. Дайте определение и перечислите основные характеристики системного программного обеспечения.
67. Дайте определение и перечислите основные характеристики прикладного программного обеспечения.
68. Дайте определение и охарактеризуйте инструментарий технологии программирования.
69. Расскажите об особенностях создания программного продукта.
70. Что такое жизненный цикл программного обеспечения?
71. Каковы основные свойства каскадной (итерационной) модели жизненного цикла?
72. Из каких этапов состоит модель жизненного цикла UML?

73. Какова стоимость исправления ошибок в ПО на различных стадиях его разработки?
74. Что такое «управление требованиями»?
75. В чем заключается анализ проблемы?
76. Какие виды ограничений на создаваемое ПО необходимо выявить в процессе работы над требованиями?
77. Каковы существующие методы выявления требований к ПО?
78. Приведите эксплуатационные требования к ПО.
79. Перечислите функциональные требования к ПО.
80. Чем определяется выбор архитектуры ПО?
81. Охарактеризуйте статические и полустатические структуры данных.
82. Охарактеризуйте динамические структуры данных.
83. Приведите понятие модуля. Характеристики модуля.
84.Какие существуют методы разработки модулей?
85. Что такое спецификации процессов?
86. Приведите пример диаграммы переходов состояний.
87. Какие бывают функциональные диаграммы?
88. Приведите пример диаграммы потоков данных.
89. Что такое диаграммы «сущность—связь»?
90. Охарактеризуйте понятие UML.
91. Опишите варианты использования системы.
92. Чем описывается поведение системы?
93.Приведите пример структурной схемы ПО.
94. Опишите основные элементы функциональных схем ПО.
95. Охарактеризуйте метод пошаговой детализации.
96. Как составляются структурные карты Константайна?
97. Как составляются структурные карты Джексона?
98. Что такое CASE-технологии?
99. Что такое RAD-технологии?
100. Охарактеризуйте модель проектируемого ПО при объектном подходе.
101. Что такое экстремальное программирование?
102. Какие виды ошибок существуют?
103. Что такое тест? Какими свойствами должен обладать тест?
104. Каковы критерии выбора тестов?
105. Дайте краткую характеристику каждому критерию выбора теста.
106. Опишите последовательность разработки тестов.
107. Что входит в понятие надежности ПО?
108. Какие виды отказов существуют?
109. Каковы количественные характеристики надежности программ?
110. Что представляют собой методы оценки и измерения характеристик надежности ПО?
111. Перечислите достоинства парного программирования.
112. Назовите виды программных документов.
113. Как составляется пояснительная записка?
114. Каким образом составляется руководство пользователя?
115. Как составляется руководство системного администратора?
116. Назовите известные вам языки программирования и перечислите их особенности.
117. Как происходит выбор среды программирования?
118. Сравните технологию .NET с технологией Java.
119. Перечислите основные сведения о защите программных продуктов.
120. Назовите криптографические средства защиты.

121. Дайте определение программным средствам защиты.
122. Какие правовые средства защиты ПП вам известны?
123. Что такое коллективная разработка ПО?
124. Что такое система контроля версий?
125. Расскажите об основных особенностях известных вам систем контроля версий.
126. Перечислите методы оценки стоимости ПО.
127. Опишите линейный метод.
128. Опишите метод функциональных точек.
129. Какие существуют модификации метода функциональных точек?
130. Приведите методы оценки стоимости ПО с использованием эмпирических данных.
131. Охарактеризуйте COCOMO и COCOMO II.
132. Как производится оценка эффективности ПО на этапе эксплуатации?
133. Что такое показатели TCO и ROI?

Критерии и шкала оценивания по оценочному средству собеседование (устный или письменный опрос)

Шкала оценивания (интервал баллов)	Критерий оценивания
5	собеседование (устный или письменный опрос) прошел на высоком уровне (студент в полном объеме осветил рассматриваемый вопрос, владеет профильным понятийным (категориальным) аппаратом и т.п.)
4	собеседование (устный или письменный опрос) прошел на среднем уровне (студент в целом осветил рассматриваемый вопрос, владеет профильным понятийным (категориальным) аппаратом и т.п.)
3	собеседование (устный или письменный опрос) на низком уровне (студент допустил существенные неточности, изложил материал с ошибками, не владеет в достаточной степени профильным категориальным аппаратом и т.п.)
2	собеседование (устный или письменный опрос) прошел на неудовлетворительном уровне или не представлен (студент не готов, не выполнил задание и т.п.)

Задания к контрольным работам

1. Создайте проект, в котором разрабатываемый класс должен содержать следующие элементы: скрытые и открытые поля, конструкторы без параметров и с параметрами (имена некоторых полей должны совпадать с идентификаторами параметров), методы и свойства. Методы и свойства должны обеспечивать непротиворечивый и удобный интерфейс класса. В программе должна выполняться проверка всех разработанных элементов класса, вывод состояния объекта.

2. Создайте проект, в котором разрабатываемый класс должен содержать следующие элементы: скрытые и открытые поля, конструкторы с параметрами и без параметров, методы, свойства, индексаторы. Класс должен реализовывать следующие операции над массивами:

задание произвольной размерности массива при создании объекта;
доступ к элементу по индексам с контролем выхода за пределы массива;
вывод на экран элемента массива по заданному индексу и всего массива.
При возникновении ошибок должны выбрасываться исключения.

В программе должна выполняться проверка всех разработанных элементов класса.

3. Создайте проект, в котором опишите класс должен, содержать следующие элементы: скрытые и открытые поля, конструкторы (один из них должен передавать параметром массив), перегруженные операции.

В программе должна выполняться проверка всех разработанных элементов класса.

4. Составить программу с одним родительским классом и потомком. Все поля должны быть закрытыми. Базовый класс должен содержать конструкторы с параметрами, методы доступа к закрытым полям, вывод полей и указанный в таблице метод. Производный класс содержит дополнения и изменения, организовать вывод новых полей потомка, при этом имена методов совпадают с именами методов базового класса. Составить тестирующую программу с выдачей результатов. Создать объекты базового и производного типов. В программе должна выполняться проверка всех разработанных элементов класса.

5. Составить программу с одним родительским классом и потомком. Все поля должны быть закрытыми. Базовый класс должен содержать конструкторы с параметрами, методы доступа к закрытым полям, вывод полей и указанный в таблице метод. Производный класс содержит дополнения и изменения, организовать вывод новых полей потомка, при этом имена методов совпадают с именами методов базового класса. Составить тестирующую программу с выдачей результатов. Создать объекты базового и производного типов. В программе должна выполняться проверка всех разработанных элементов класса.

6. Составить программу с одним родительским классом и двумя потомками. Потомки должны содержать виртуальные функции. Создать виртуальную функцию выдачи результатов расчета методов на экран монитора с указанием названий и полей и их значений соответствующего объекта. Составить тестирующую программу с выдачей протокола на экран монитора. При этом создать объекты базового и производных типов, используя полиморфный контейнер - массив ссылок базового класса на объекты базового и производных классов (количество объектов >=5).

7. Составить программу с абстрактным родительским классом и двумя объектами - потомками. Для этого модифицировать задание 1. Составить тестирующую программу с выдачей протокола на экран монитора. В ней нужно реализовать циклический вывод параметров объектов, используя полиморфный контейнер - массив объектов базового класса (количество объектов >=5).

8. Интерфейсы Ix, Iy, Iz, содержат объявления методов с одной и той же сигнатурой следующим образом

```
interface Ix
{
void IxF0(параметр);
void IxF1();
}
interface Iy
{
void F0(параметр);
void F1();
}
interface Iz
{
void F0(параметр);
```

void F1();
}

Эти интерфейсы наследуются в классе TestClass, содержащий член w типа параметр. В каждом методе задать вывод результата.

Рассмотреть случай:
неявной реализации интерфейсов
явной реализации интерфейса Iz
В программе должна выполняться:
неявная неоднозначная реализация методов интерфейсов Iy и Iz,
вызов функций с явным приведением к типу интерфейса,
вызов метода для объекта посредством интерфейсной ссылки.

9. Выполнить задания, используя для хранения экземпляров разработанных классов стандартные параметризованные коллекции. Во всех классах реализовать интерфейсы IComparable и IComparer перегрузить операции отношения для реализации сравнения объектов по указанному полю. Результат вывести на экран.

Критерии и шкала оценивания по оценочному средству «контрольная работа»

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Контрольная работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Контрольная работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Контрольная работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов/задач)
2	Контрольная работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Оценочные средства для промежуточной аттестации (экзамен)
Теоретические вопросы

1. Технология программирования в историческом аспекте. Основные понятия и определения.
2. Классификация программного обеспечения.
3. Системное программное обеспечение.
4. Инструментарий технологии программирования.
5. Пакеты прикладных программ.
6. Принципы работы с требованиями к программному обеспечению. Проблематика проектирования. Оценка стоимости ошибок.
7. Управление требованиями. Последовательность работы с требованиями. Анализ проблемы. Преграды на пути выявления требований.
8. Серия стандартов ISO 9000. CMM. Процесс сертификации программ на базе информации об их использовании.
9. Модели жизненного цикла.
10. Rational Objectory Process — модель жизненного цикла (методология объектно-ориентированного программиро-вания).
11. Жизненный цикл UML (Rational Objectory Process).
12. Специфицирование и планирование. Процесс разработки. Выпуск продукта и механизмы обратной связи.
13. Функциональные требования.
14. Эксплуатационные требования.

15. Выбор архитектуры программного обеспечения.
16. Структура и формат данных.
17. Статические, полустатические и динамические структуры.
18. Понятие модуля. Основные характеристики программного модуля.
19. Модульная структура программных продуктов.
20. Методы разработки при модульном программировании.
21. Спецификации процессов.
22. Словарь терминов.
23. Диаграммы переходов состояний (SDT).
24. Функциональные диаграммы.
25. Диаграммы потоков данных (DFD).
26. Диаграммы сущность—связь.
27. Определение прецедентов (вариантов использования).
28. Построение концептуальной модели предметной области.
29. Описание поведения системы.
30. Диаграммы последовательностей, деятельности и состояний.
31. Структурная схема разрабатываемого программного обеспечения.
32. Функциональная схема.
33. Метод пошаговой детализации при составлении алгоритмов.
34. Структурные карты Константайна.
35. Структурные карты Джексона.
36. CASE-технологии.
37. Ускорение разработки программного обеспечения. Методология RAD.
38. Разработка структуры программного обеспечения при объектном подходе. Диаграммы кооперации.
39. Основополагающие практики XP. Преимущества простого дизайна. Простой дизайн.
40. Рефакторинг и принцип YAGNI. Наращивание архитектуры. UML и XP. Суть проектирования. Программирование и тестирование.
41. Тестирование «белого ящика» и «черного ящика».
42. Порядок разработки тестов.
43. Автоматизация тестирования.
44. Модульное тестирование.
45. Интеграционное тестирование.
46. Системное тестирование. Эффективность и оптимизация.
47. Стиль программирования. Надежность программного обеспечения. Отладка программ.
48. Инструментальные средства разработки программ. Защита программных продуктов.
49. Виды программных документов.
50. Пояснительная записка.
51. Руководство пользователя.
52. Руководство системного программиста. Коллективная разработка программного обеспечения.
53. Оценка стоимости разработки программного обеспечения.
54. Методы оценки эффективности ПО на этапе эксплуатации.

Практические задания

1. Описать класс для работы с одномерным массивом целых чисел (вектором). Класс должен реализовывать возможность: выполнение операции нахождения остатков от деления всех элементов массива на скаляр.

2. Описать класс для работы с одномерными массивами чисел. Класс должен реализовывать возможность: выполнения для массивов комбинированных операций присваивания (+=, -=).

Описать класс для работы с одномерным массивом строк фиксированной длины. Обеспечить следующие возможности: сравнения массивов на равенство (перегрузку операции == для поэлементного сравнения строк).

4. Описать класс, реализующий тип данных «вещественная матрица». Класс должен реализовывать следующие операции над матрицами: вычитание заданной номером строки из всех остальных строк, кроме данной строки.

5. Описать класс для работы с одномерным массивом строк фиксированной длины. Обеспечить следующие возможности: перегрузку операции + для поэлементного соединения массивов.

6. Описать класс для работы с n-мерным вектором. Класс должен реализовывать возможность: перегруженные операции отношений, выполняющие сравнение длин векторов;

7. Описать класс для работы с одномерным массивом чисел. Класс должен реализовывать возможность: выполнение операций поэлементного умножения массивов с одинаковыми границами индексов.

8. Описать класс для работы с одномерным массивом целых чисел (вектором). Класс должен реализовывать возможность: уменьшение количества элементов массива на заданное число (перегрузка операции -).

9. Описать класс для работы с одномерным массивом вещественных чисел. Обеспечить следующие возможности: нахождение суммы элементов массива (перегрузка операции +).

10.Описать класс, реализующий тип данных «вещественная матрица». Класс должен реализовывать следующие операции над матрицами: изменение значений элементов матрицы на противоположные.

11.Описать класс для работы с одномерным массивом целых чисел. Класс должен реализовывать возможность: нахождения числа, полученного перемножением положительных элементов массива.

12.Описать класс для работы с одномерным массивом чисел, позволяющий выполнять основные операции: добавление и удаление элемента в массив – перегруженные операции ++ и --.

13. Описать класс, реализующий тип данных «вещественный массив» и работу с ним. Класс должен реализовывать метод, проверяющий является ли матрица симметричной.

14. Описать класс для работы с двумерным массивом символов, состоящих из одних цифр. Обеспечить следующие возможности: рассматривая символы как числа, определить сумму четных и нечетных цифр в каждой строке.

15. Описать класс для работы с одномерным массивом целых чисел. Обеспечить возможность нахождения суммы элементов, стоящих после введенного с клавиатуры значения и вывода элементов, стоящих после введенного с клавиатуры значения.

16. Описать класс «список», состоящий из номеров зачетной книжки и годов рождений студентов. Класс должен реализовывать метод, вычисляющий, сколько лет студенту и вывод на экран информации вида «номер зачетной книжки – количество лет».

17. Описать класс, реализующий тип данных «матрица». Класс должен реализовывать возможность нахождения количества столбцов, начинающихся с отрицательного числа.

18. Описать класс, реализующий тип данных «матрица целых чисел». Класс должен реализовывать метод проверки, является ли матрицы верхней треугольной.

19. Описать класс, реализующий тип данных «вещественная матрица». Класс должен реализовывать возможность преобразования матрицы следующим образом: каждый элемент строки разделить на максимальный элемент этой строки, если он не равна 0. В противном случае элементы строки оставить без изменений.

20. Описать класс для работы с одномерным массивом вещественных чисел. Предусмотреть возможность добавления элемента в массив по заданному индексу.

21. Описать класс для работы с двумерным числовым массивом. Обеспечить возможность проверки, является ли элемент массива палиндромом. Палиндром принимает одно и то же значение при чтении его как справа налево, так и слева направо.

22.Описать класс для работы с двумерным массивом целых чисел. Предусмотреть возможность поиска элемента в массиве по заданному значению.

23.Описать класс для работы с одномерным массивом вещественных чисел. Предусмотреть возможность удаления элемента из массива по заданному индексу.

24.Описать класс для работы с одномерным массивом целых чисел. Обеспечить возможность определения, является ли массив упорядоченным.

Критерии и шкала оценивания к промежуточной аттестации «экзамен»

Национальная шкала	Характеристика знания предмета и ответов
отлично (5)	Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы

9. Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов

При необходимости рабочая программа учебной дисциплины может быть адаптирована для обеспечения образовательного процесса инвалидов и лиц с ограниченными возможностями здоровья, в том числе с применением электронного обучения и дистанционных образовательных технологий.

Для этого требуется заявление студента (его законного представителя) и заключение психолого-медико-педагогической комиссии (ПМПК). В случае необходимости обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося), а для инвалидов также в соответствии с индивидуальной программой

реабилитации инвалида могут предлагаться следующие варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей:

– создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;

– создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;

– создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников, например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения – аудиально;

– применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной (модулем), за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;

– применение электронного обучения, дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;

– применение электронного обучения, дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;

– увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:

– продолжительность сдачи зачёта или экзамена, проводимого в письменной форме, – не более чем на 90 минут;

– продолжительность подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, – не более чем на 20 минут; – продолжительность выступления обучающегося при защите курсовой работы – не более чем на 15 минут.

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)