

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
(КГУ)

Кафедра «Программное обеспечение автоматизированных систем»



УТВЕРЖДАЮ:

Врио ректора

/Н.В. Дубив/

«*ОК*» *сентябрь* 20 *19* г.

Рабочая программа учебной дисциплины
ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

образовательной программы высшего образования –
программы специалитета

10.05.03 – Информационная безопасность автоматизированных систем

Направленность:

*Обеспечение информационной безопасности
распределенных информационных систем*

очная форма обучения

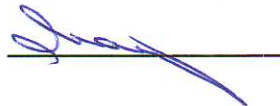
Курган 2019

Программа дисциплины «Технологии и методы программирования» составлена в соответствии с учебным планом программы специалитета 10.05.03 – Информационная безопасность автоматизированных систем (Обеспечение информационной безопасности распределенных информационных систем), утвержденным 29 августа 2019 года.

Рабочая программа одобрена на заседании кафедры «Программное обеспечение автоматизированных систем» «30» августа 2019 года, протокол № 1.

Рабочую программу разработали:

Разделы 1 – 3, курсовая работа:
доцент кафедры ПОАС



А.В. Маер

Раздел 4:
доцент кафедры ПОАС



В.К. Волк

Согласовано:

Заведующий
кафедрой ПОАС



Т.Р. Змызгова

Заведующий
кафедрой БИАС



Е.Н. Полякова

Начальник
Управления
образовательной деятельности



С.Н. Синицын

Специалист
по учебно-методической работе
Учебно-методического отдела



Г.В. Казанкова

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 6 зачетных единиц (216 академических часа)

Виды учебной работы	Распределение трудоемкости по семестрам и видам учебных занятий, акад. часов		
	Всего	3-й семестр	4-й семестр
Общая трудоемкость дисциплины	216	108	108
Аудиторные занятия, всего:	80	48	32
Лекции	48	32	16
Лабораторные работы	32	16	16
Самостоятельная работа, всего:	136	60	76
Подготовка к экзамену	54	27	27
Контрольная работа	18	18	-
Курсовая работа (КР)	36	-	36
Другие виды самостоятельной работы	28	15	13
Виды промежуточной аттестации	Два экзамена, Защита КР	Экзамен	Экзамен, Защита КР

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Технологии и методы программирования» относится к Блоку 1 базовой части образовательной программы (дисциплины, модули), для ее освоения необходимы компетенции, сформированные при изучении дисциплин «Информатика», «Языки программирования» и «Дискретная математика».

Результаты изучения дисциплины необходимы для разработки и программной реализации алгоритмов решения прикладных задач обработки информации и моделирования процессов и объектов различной естественно-научной природы и для освоения технологий проектирования информационных систем различного назначения.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Цель изучения дисциплины:

– повышение уровня профессиональной подготовки студентов в области алгоритмизации и программирования, а также ознакомление с технологиями и стандартами промышленной разработки программного обеспечения (далее - ПО).

Задачами дисциплины являются изучение:

– методов организации линейных, иерархических и сетевых структур данных и типовых алгоритмов сортировки и поиска данных, реализуемых на этих структурах;

– получение практических навыков программирования задач обработки данных повышенной сложности и анализа временных характеристик алгоритмов;

– ознакомление с основными понятиями, методическими основами и стандартами программной инженерии;

– ознакомление с основными моделями жизненного цикла ПО, промышленными технологиями его разработки и инструментальными средствами.

Компетенции, формируемые в результате освоения дисциплины:

- способность применять языки, системы и инструментальные средства программирования в профессиональной деятельности (ОПК-3);
- способность к освоению новых образцов программных, технических средств и информационных технологий (ОПК-8);
- способность разрабатывать и анализировать проектные решения по обеспечению безопасности автоматизированных систем (ПК-8);
- способностью применять знания в области электроники и схемотехники, технологий, методов и языков программирования, технологий связи и

передачи данных при разработке программно-аппаратных компонентов защищенных автоматизированных систем в сфере профессиональной деятельности (ПК-10).

В результате изучения дисциплины обучающийся

Должен знать:

- языки, системы и инструментальные средства программирования в профессиональной деятельности (для ОПК-3);
- новые образцы программных, технических средств и информационных технологий (для ОПК -8);
- проектные решения по обеспечению безопасности автоматизированных систем (для ПК-8);
- технологии, методы и языки программирования, используемые при разработке программно-аппаратных компонентов защищенных автоматизированных систем в сфере профессиональной деятельности (для ПК-10).

Должен уметь:

- применять языки, системы и инструментальные средства программирования в профессиональной деятельности (для ОПК -3);
- осваивать новые образцы программных, технических средств и информационных технологий (для ОПК -8);
- разрабатывать и анализировать проектные решения по обеспечению безопасности автоматизированных систем (для ПК-8);
- применять знания в области электроники и схемотехники, технологий, методов и языков программирования при разработке программно-аппаратных компонентов защищенных автоматизированных систем в сфере профессиональной деятельности (для ПК-10).

4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Учебно-тематический план

Разделы дисциплины		Часов контактной работы с преподавателем	
№	Наименование	Лекции	Лаб. работы
3-й семестр			
1	Типы и структуры данных в высокоуровневых языках программирования	8	4
2	Алгоритмы сортировки данных	8	4
Рубежный контроль №1		-	1
3	Алгоритмы поиска данных	16	6
Рубежный контроль №2			1
4-й семестр			
4	Технологии промышленной разработки ПО	16	12
Рубежный контроль №3		-	2
Рубежный контроль №4		-	2
Всего по дисциплине:		48	32

4.2 Курс лекций

Номер раздела	Наименование раздела, краткое содержание лекции	Часов контактной работы с преподавателем
		Лекции
3-й семестр		
1	<p style="text-align: center;">РАЗДЕЛ 1. Типы и структуры данных в высокоуровневых языках программирования</p> <p><i>Введение:</i> обзор программы изучения дисциплины; цели, задачи, структура дисциплин; мероприятия текущего, рубежного и итогового контроля; учебная литература и информационные источники. <i>Концепция типа данных</i> в языках программирования высокого уровня. Классификация типов данных. Базовые и конструируемые типы. Скаляры, массивы, записи, множества. Статические и динамические типы. Указатели. Представление и реализация типов данных. <i>Линейные структуры данных.</i> Списки: односвязные, двусвязные, циклические. Мультисписки. Очереди, стеки, деки. Способы программной реализации и методы обработки линейных структур данных.</p> <p>Определение, базовые элементы и характеристики деревьев. Степень (порядок) и глубина (высота) дерева. Бинарные и сильноветвящиеся деревья. Мультисписковое представление деревьев. Регулярные и нерегулярные деревья. Идеально сбалансированные деревья и деревья, сбалансиро-</p>	8

Номер раздела	Наименование раздела, краткое содержание лекции	Часов контактной работы с преподавателем
		Лекции
	<p>ванные по высоте (АВЛ-деревья). Методы поддержки сбалансированности дерева при вставке и удалении узлов. Классические В-деревья и В+-деревья.</p> <p>Определение, базовые элементы и характеристики графов. Понятия "смежности", "инцидентности", "степени" графа. Ориентированные и неориентированные графы. Пути и маршруты на графе. Списковое и матричное и представление графов. Матрицы смежности, матрицы инцидентности.</p>	
2	<p>РАЗДЕЛ 2. Алгоритмы сортировки данных</p> <p>Сортировка записей в массиве как способ ускорения поиска по значению ключевого поля. Метод дихотомии. Постановка задачи сортировки данных в оперативной памяти. Сортировка включением. Сортировка обменом (метод "пузырька") и его модификации. Сортировка выбором. Сортировка разделением (QuickSort). Методы сортировки, основанные на деревьях. Пирамидальная сортировка. Сортировка слиянием. Оценка временной сложности программной реализации алгоритмов. "Big-O" – нотация. Сравнительные оценки методов.</p> <p>Постановка задачи сортировки данных, размещенных в последовательных файлах на внешних запоминающих устройствах. Методы сортировки слиянием: прямое слияние, естественное слияние, сбалансированное многопутевое слияние. Оценка временной сложности методов</p> <p>Рубежный контроль 1</p>	8
3	<p>Раздел 3. Алгоритмы поиска данных</p> <p>Постановка задачи поиска. Алгоритм прямого поиска, алгоритм Кнута, Морриса и Пратта (КМП), алгоритм Бойера-Мура. Сравнительная оценка временной сложности алгоритмов.</p> <p>Задача поиска записи по значению её ключевого поля. Первичные и вторичные ключи. Построение деревьев поиска. Алгоритмы прохождения дерева. Деревья оптимального поиска. Деревья цифрового поиска.</p> <p><i>Поиск данных во внешних файлах.</i> Индексные структуры на основе В-деревьев и В+-деревьев. Порядок и глубина (высота) дерева. Фактор заполнения. Индексно-последовательные файлы ("кластерные" индексы). Организация индексов по первичным и вторичным ключам. Композитные индексы.</p> <p><i>Методы обхода графов.</i> Задача анализа структуры графа и вычисления его метрических характеристик. Стратегии обхода графа: поиск в ширину (BFS) и поиск в глубину (DFS). Алгоритмы реализации процедур обхода. Сравнительная оценка временной сложности алгоритмов обхода графов. <i>Алгоритмы поиска кратчайших путей.</i> Постановка задачи поиска кратчайшего пути на графе. Алгоритмы Дейкстры, Флойда и Йена. Оценка сложности алгоритмов поиска кратчайших путей.</p> <p>Рубежный контроль 2</p>	16
4-й семестр		

Номер раздела	Наименование раздела, краткое содержание лекции	Часов контактной работы с преподавателем
		Лекции
4	<p>Раздел 4. Технологии промышленной разработки ПО</p> <p><i>Предмет и базовые понятия программной инженерии.</i></p> <p>Проблема увеличения сложности, стоимости и сроков разработки ПО. История развития технологий разработки программных продуктов. Понятие инженерной деятельности. Инженеры и программные инженеры. Понятие и ролевая модель команды программного проекта. Обзор профессиональных стандартов в IT-области: профессии IT-специалистов; функции и должностные обязанности; требования к квалификационным уровням и профессиональной компетентности.</p>	2
	<p><i>Процессы и модели жизненного цикла ПО.</i></p> <p>Понятие <i>жизненного цикла</i> (ЖЦ) промышленного изделия, как последовательности взаимосвязанных процессов, реализуемых на всех стадиях его существования. Проектирование как процесс преобразования информационных моделей создаваемого объекта. Декомпозиция, многоэтапность и итерационность как базовые методы "борьбы со сложностью" проекта. Типовые стадии проекта. Проектная документация.</p> <p><i>Жизненный цикл программного продукта.</i> Обзор стандарта ISO/IEC 12207. Классификация и иерархия процессов ЖЦ ПО: основные, вспомогательные и организационные процессы. Участники процессов ЖЦ ПО. Основные процессы ЖЦ ПО: "заказ", "поставка", "разработка", "эксплуатация", "сопровождение". Типовая структура процесса "разработка": подготовка процесса; анализ требований к системе; проектирование системной архитектуры; анализ требований к программным модулям; проектирование программной архитектуры; техническое проектирование программных модулей; программирование, тестирование, сборка и квалификационные испытания программных модулей; сборка системы; квалификационные испытания системы; ввод в действие и обеспечение приемки.</p> <p><i>Модели жизненного цикла ПО.</i> Понятие модели ЖЦ ПО. Обзор типовых моделей ЖЦ ПО (каскадная, эволюционная и спиральная модели). Промышленные технологии создания ПО: технология <i>Rational Unified Process (RUP)</i>; технология <i>Microsoft Solution Framework (MSF)</i>, технология <i>eXtreme Programming (XP)</i>.</p>	4
	Рубежный контроль №3	

Номер раздела	Наименование раздела, краткое содержание лекции	Часов контактной работы с преподавателем
		Лекции
	<p><i>Визуальное моделирование при анализе и проектировании программных систем.</i></p> <p>Проектирование как процесс преобразования информационных моделей объекта. Задачи и базовые принципы проектирования сложных объектов (абстрагирование, декомпозиция, многоэтапность, итерационность). Визуализация при проектировании. Обзор систем графического моделирования. Структурные модели анализа бизнес-процессов: схема Захмана. Диаграммы структурного анализа систем: SADT - диаграммы функционального моделирования; ERD - диаграммы "Сущность – Связь"; DFD - диаграммы потоков данных. ООАП - объектно-ориентированный подход к анализу и проектированию программных систем. Язык графического моделирования UML: история стандартизации UML; структура и базовые понятия языка; модели и UML-диаграммы.</p>	4
	<p><i>UML-диаграммы вариантов использования (UseCase)</i></p> <p>Use Case-модель как результат функциональной декомпозиции проектируемой системы на начальной стадии программного проекта. Назначение и область применения Use-Case-моделей. Компоненты модели и графическая нотация. Сценарии вариантов использования. Примеры UseCase-диаграмм.</p>	2
	<p><i>UML-диаграммы классов и пакетов (Class- и Package-diagram)</i></p> <p>Статические модели концептуального и логического уровней. «Пакеты» и «Классы» – два уровня структурной декомпозиции проектируемой системы. Компоненты моделей (пакеты, классы, интерфейсы, отношения), графическая нотация. Примеры диаграмм.</p>	2
	<p><i>UML-диаграммы деятельности и диаграммы состояний (Activity-diagram и State Machine-diagram)</i></p> <p>Назначение и область применения динамических UML-моделей. Модели логического уровня. Компоненты модели: простые и составные состояния; события и переходы; триггеры. Обозначения и графическая нотация. Примеры диаграмм.</p>	2
	Рубежный контроль №4	
Всего часов по дисциплине:		48

4.3 Лабораторный практикум

Номер лаб. работы	Наименования разделов и лабораторных работ	Часов контактной работы с преподавателем
3-й семестр		
Раздел 1. Типы и структуры данных в высокоуровневых языках программирования		
1	<i>Типовые алгоритмы обработки линейных структур данных</i>	4
Раздел 2. Алгоритмы сортировки данных		
2	<i>Алгоритмы сортировки данных в основной памяти</i>	2
3	<i>Алгоритмы сортировки данных во внешней памяти</i>	2
Рубежный контроль №1		1
Раздел 3. Алгоритмы поиска данных		
4	<i>Алгоритмы поиска подстроки в строке</i>	3
5	<i>Алгоритмы обработки иерархических структур данных</i>	3
Рубежный контроль №2		1
4-й семестр		
Раздел 4. Технологии промышленной разработки ПО		
Рубежный контроль №3		2
6	<i>Подготовка к выполнению учебного программного проекта</i> Рассмотрение примера выполнения программного проекта. Выбор варианта темы проекта. Постановка задачи разработки.	2
7	<i>Разработка обобщенной UseCase-модели</i> <u>Стадия технического задания</u> : формирование терминологического словаря предметной области; разработка обобщенной диаграммы вариантов использования.	2
8	<i>Разработка диаграммы пакетов, UseCase-моделей и сценариев вариантов использования</i> <u>Стадия эскизного проекта</u> : структурная декомпозиция проектируемой системы (разработка диаграммы пакетов); функциональная декомпозиция подсистем (разработка локальных Use Case-диаграмм и сценариев вариантов использования)	4
9	<i>Разработка диаграммы классов</i> <u>Стадия технического проекта</u> – разработка статической модели проектируемой системы (компонента системы): структурная декомпозиция пакетов, разработка диаграмм классов.	2
10	<i>Разработка диаграммы состояний</i> <u>Стадия технического проекта</u> – разработка динамической модели поведения проектируемой системы (компонента системы): разработка диаграммы состояний (<i>StateMachine-diagram</i>).	2
Рубежный контроль №4		2
Всего часов лабораторных занятий		32

4.4 Контрольная работа

Контрольная работа выполняется в 3-м семестре по материалу первых трех разделов дисциплины. Варианты заданий и требования к результатам выполнения контрольной работы приводятся в методических указаниях.

Примерный вариант задания:

1. Докажите, что куча из n элементов имеет высоту $\lfloor \lg n \rfloor$.
2. Измените процедуру *HEAPIFY*, заменив рекурсию циклом.
3. Как правило, в банках обрабатывают чеки в порядке их поступления; клиенты же предпочитают, чтобы в отчете платежи были указаны в порядке номеров чеков. Владелец чековой книжки обычно выписывает чеки подряд, а получатели чеков предъявляют их в банк вскоре после выписывания. Таким образом, порядок номеров нарушается незначительно. Следовательно, банку требуется отсортировать почти отсортированный массив. Объясните, почему сортировка вставками в таких случаях работает быстрее, чем быстрая сортировка.

4.5 Курсовая работа

Курсовая работа выполняется в 4-м семестре по материалу первых трех разделов дисциплины. Варианты заданий и рекомендации по выполнению курсовой работы приведены в методических указаниях.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

5.1 Курс лекций

Лекционный курс базируется на пассивном методе обучения, реализующем традиционную объяснительно-иллюстративную образовательную технологию, в рамках которой студенты выступают в роли слушателей, воспринимающих учебный материал и участвующих в дискуссиях и экспресс-опросах.

Конспект лекций (краткий обзор рассматриваемых на лекциях вопросов) представлен в формате мультимедийных презентаций и включен в состав учебно-методического комплекса дисциплины, доступного студентам. Более детальное содержание лекционного материала представлено в соответствующих учебных пособиях, структура которых соответствует тематическому плану изучения дисциплины. Учебные пособия содержат перечни контрольных вопросов, ответы на которые должны быть получены студентами в процессе самостоятельной проработки материала соответствующей лекции.

5.2 Лабораторный практикум

Лабораторные занятия проводятся на основе интерактивных методов в виде творческих заданий экспериментального характера, направленных не

столько на закрепление уже изученного материала, сколько на изучение нового, и выполняемые студентами, объединяемыми в малые группы (2-3 человека). Задания не имеют однозначного решения и соответствуют целям обучения.

Краткое теоретическое введение по каждой из лабораторных работ, практические задания и методические указания к их выполнению, а также требования к оформлению отчетов, приведены в соответствующих методических указаниях по их выполнению.

При выполнении лабораторных работ рекомендуется использование технологий развивающейся кооперации, коллективного взаимодействия, разбора конкретных ситуаций.

Лабораторные работы №1 – №5 предполагают программную реализацию определенного алгоритма и выполнение небольшого экспериментального исследования, на основе разработанной программы.

Лабораторные работы №6 – №10 объединены общей темой учебного программного проекта, в рамках которого последовательно разрабатываются соответствующие UML-диаграммы. Каждая лабораторная работа реализует одну из стадий программного проекта и требует освоения и применения соответствующих CASE-средств. Защита проектов проводится в форме собеседования по материалу представленного отчета и сделанного публичного доклада. В процессе защиты оценивается полнота и качество выполнения практических заданий каждым из участников команды проекта, грамотность использования инструментальных средств, качество оформления отчета.

5.3 Самостоятельная работа

Самостоятельная работа по освоению дисциплины включает изучение теоретических разделов дисциплины, подготовку к лекционным занятиям, выполнение лабораторных работ и контрольной работы, выполнение и защиту курсовой работы, подготовку к рубежному контролю и промежуточной аттестации.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:

Виды самостоятельной работы	Рекомендуемая трудоемкость, акад. часов
<i>Изучение материала лекционного курса:</i>	6
РАЗДЕЛ 1. Типы и структуры данных в высокоуровневых языках программирования	1
РАЗДЕЛ 2. Алгоритмы сортировки данных	1
РАЗДЕЛ 3. Алгоритмы поиска данных	2
РАЗДЕЛ 4. Технологии промышленной разработки ПО	2
<i>Подготовка и выполнение лабораторных работ (по 2 часа на каждое занятие)</i>	14
<i>Подготовка к рубежному контролю (по 2 часа на каждый рубеж)</i>	8
<i>Выполнение курсовой работы</i>	36
<i>Выполнение контрольной работы</i>	18

<i>Подготовка к экзаменам</i>	54
Всего:	136

6. КОНТРОЛЬ И АТТЕСТАЦИЯ

6.1 Состав и формы проведения контроля и аттестации

Программой изучения дисциплины предусмотрены мероприятия текущего и рубежного контроля и промежуточная аттестация (в форме экзамена).

График и формы проведения контрольных и аттестационных мероприятий приведены в таблице:

Виды	Содержание	Форма проведения	Неделя
3-й семестр			
Текущий контроль	Контроль посещения лекционных занятий	-	1 – 16
	Контроль выполнения лабораторных работ	Демонстрация программной реализации, собеседование	4 – 16
	Выполнение контрольной работы	Защита отчета, собеседование	15
Рубежный контроль	№1. Алгоритмы сортировки данных	Тестирование	6
	№2. Алгоритмы поиска данных	Тестирование	15
Промежуточная аттестация	Экзамен	Собеседование	-
4-й семестр			
Текущий контроль	Контроль посещения лекционных занятий	-	1 – 16
	Контроль выполнения лабораторных работ	Демонстрация результатов выполнения этапов учебного программного проекта, собеседование	4 – 16
Рубежный контроль	№3. Стандарты и модели жизненного цикла ПО	Тестирование	6
	№4. Визуальное моделирование при проектировании ПО	Тестирование	15
Промежуточная аттестация	Экзамен	Собеседование	-

6.2 Процедура оценивания результатов освоения дисциплины

6.2.2 Текущий контроль

Контроль выполнения контрольной и лабораторных работ проводится в форме защиты отчетов с демонстрацией полученных результатов.

Оценивается своевременность и качество выполнения заданий, качество программного кода/проектных решений, а также эрудиция студента в вопросах, связанных с тематикой выполненной работы.

6.2.2 Рубежный контроль

Рубежный контроль проводится в форме фронтального тестирования по теоретической части дисциплины. Оценивается количество правильных ответов на задания теста и соответственно начисляется балл. Студент, ответивший правильно менее, чем на 50% заданий теста, считается не прошедшим тестирование и обязан повторно пройти этот тест во время консультации по дисциплине.

6.2.3 Промежуточная аттестация

Экзамены по дисциплине проводятся в традиционной (устной) форме. Экзаменационный билет содержит 2 теоретических вопроса и одну задачу. На подготовку ответа студенту отводится один астрономический час. Оценивается качество ответа и решения задачи (максимально 10 баллов за ответ на один вопрос или задачу).

6.3 Система балльно-рейтинговой оценки работы студентов

Оценивание результатов выполнения студентами плановых контрольных и аттестационных мероприятий по дисциплине производится в соответствии с Положением о балльно-рейтинговой системе контроля и оценки академической активности студентов ФГБОУ ВО «Курганский государственный университет».

6.3.1 Критерии оценивания

Оценивание производится по 100-балльной (за семестр) шкале с последующим приведением итоговой 100-балльной рейтинговой оценки к традиционной четырех-балльной.

Рейтинговая оценка студента получается путем суммирования баллов, полученных студентом в течение семестра (максимум 70 баллов) и баллов, полученных им на промежуточной аттестации (максимум 30 баллов).

Курсовая работа оценивается отдельно: до 70 баллов за своевременность и качество выполнения работы в течение семестра и до 30 баллов на промежуточной аттестации (на защите курсовой работы).

Минимальное количество баллов, которыми может быть оценен удовлетворительный ответ студента на экзамене или защите курсовой работы, равно 11. Неудовлетворительный ответ оценивается в 0 баллов.

Максимальные балльные оценки по результатам проведения контрольных и аттестационных мероприятий приведены в таблицах:

3-й семестр

Виды контроля / аттестации	Содержание	Максимальная оценка	
		За единицу	Всего
Текущий контроль	Контроль посещения лекционных занятий	1	16
	Выполнение лабораторных работ	5	25
	Выполнение контрольной работы	9	9
Рубежный контроль	№1. Структуры данных и алгоритмы сортировки.	10	10
	№2. Алгоритмы поиска данных	10	10
Промежуточная аттестация (экзамен)		30	30
Максимальная итоговая оценка, баллов			100

4-й семестр

Виды контроля / аттестации	Содержание	Максимальная оценка	
		За единицу	Всего
Текущий контроль	Контроль посещения лекционных занятий	1	8
	Выполнение лабораторных работ	10	50
Рубежный контроль	№3. Стандарты и модели жизненного цикла ПО	5	5
	№4. Визуальное моделирование при проектировании ПО	7	7
Промежуточная аттестация (экзамен)		30	30
Максимальная итоговая оценка, баллов			100

Курсовая работа

Виды контроля / аттестации	Содержание	Максимальная оценка	
		За единицу	Всего
Текущий контроль	Качество оформления пояснительной записки	30	30
	Качество программной реализации	40	40
Промежуточная аттестация (защита КР)		30	30
Максимальная итоговая оценка, баллов			100

Пересчет 100-балльной рейтинговой оценки студента в традиционную (4-балльную) оценку и в оценку ECTS (Общеввропейская система учета учебной работы) производится в соответствии с таблицей:

Рейтинговая оценка, баллов	Виды оценок промежуточной аттестации		
	Традиционная оценка		Оценка ECTS
91-100	5	Отлично	A
84-90	4	Очень хорошо	B
74-83		Хорошо	C
68-73	3	Удовлетворительно	D
61-67		Посредственно	E
51-60	2	Неудовлетворительно	Fx
0-50			F

6.3.2 Критерии допуска к промежуточной аттестации

Для допуска к промежуточной аттестации студент должен набрать по итогам текущего и рубежного контроля в течение семестра не менее 50 баллов и при этом он должен выполнить и защитить все лабораторные работы, контрольную работу (3-й семестр) и курсовую работу (4-й семестр).

В случае, если по результатам текущего и рубежного контроля студентом набрано менее 50 баллов, он может набрать недостающее количество баллов, выполнив дополнительные индивидуальные задания до конца зачетной недели семестра.

Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, также проводится путем выполнения дополнительных индивидуальных заданий.

Состав дополнительных заданий, их количество, формы выполнения и максимальные балльные оценки определяются преподавателем и доводятся до студента в момент выдачи заданий.

Для получения оценки «удовлетворительно» автоматически (без сдачи экзамена) студенту достаточно набрать 68 баллов по результатам текущего и рубежного контроля в течение семестра.

Студенту, набравшему в течение семестра не менее 68 баллов, преподаватель вправе добавить до 30 дополнительных (бонусных) баллов за активность на учебных занятиях, оригинальность принимаемых решений при выполнении лабораторных работ и контрольных заданий и выставляется «автоматически» оценка хорошо или отлично.

6.4 Фонд оценочных средств

Полный банк заданий для текущего и рубежных контролей, промежуточной аттестации по дисциплине, показатели, критерии шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов приведены в учебно-методическом комплексе дисциплины

6.4.1 Перечень оценочных средств

- 1) Балльно-рейтинговая система контроля и оценки академической активности студентов в КГУ.
- 2) Задания для тестирования (4 рубежных контроля).
- 3) Задания для выполнения контрольной работы.
- 4) Вопросы для подготовки к экзамену по дисциплине.
- 5) Образцы отчетов по выполнению лабораторных работ.
- 6) Образцы отчетов по выполнению контрольной работы.
- 7) Курсовая работа.

6.4.2 Примерные варианты компонентов фонда оценочных средств

6.4.2.1 Примеры вопросов для проведения экзамена

3-й семестр:

1. Концепция "типа данных" в программировании на языках высокого уровня. Категории типов данных. Встроенные (базовые) и конструируемые (составные) типы данных.
2. Статически- и динамически определяемые структуры данных. Указатели.
3. Линейные структуры данных (списки, очереди, стеки): способы представления и методы обработки.
4. Иерархические структуры данных (деревья): типы, основные характеристики, способы представления.
5. Сетевые структуры данных (графы): типы, основные характеристики, способы представления.
6. Простые алгоритмы внутренней сортировки (прямое включение, прямой обмен, прямой выбор). Модификации и оценка сложности алгоритмов.
7. Быстрые алгоритмы внутренней сортировки (на примере метода Quicksort).
8. Алгоритмы внутренней сортировки с использованием деревьев.
9. Алгоритмы внешней сортировки прямым и многопутевым слиянием.
10. Алгоритмы линейного и двоичного поиска.
11. Поиск в строках: алгоритм прямого поиска подстроки; КМП-алгоритм.
12. Алгоритмы поиска, удаления и вставки ключей на бинарных деревьях.
13. Использование классических В-деревьев для поиска ключей в файлах данных.
14. Алгоритмы удаления, вставки и модификации ключей в классических В-деревьях.
15. Использование В+-деревьев для поиска ключей в файлах данных.
16. Алгоритмы удаления, вставки и модификации ключей в В+-деревьях.
17. Методы регулярного обхода графа в задачах анализа его структуры и метрических характеристик. Процедура "поиска в ширину".
18. Методы регулярного обхода графа в задачах анализа его структуры и метрических характеристик. Процедура "поиска в глубину".
19. Алгоритмы поиска кратчайших путей на графах (на примере алгоритмов Дейкстры и Флойда).

4-й семестр:

1. Понятия "*программный продукт*" и "*программное обеспечение*" (software). "*Коробочные*" и "*заказные*" программные продукты. Концепции программной инженерии как промышленной технологии разработки и сопровождения ПО.

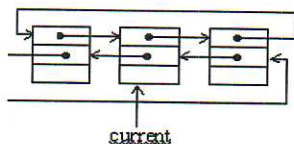
2. Понятие "жизненного цикла ПО". Типовая ролевая модель команды программного проекта. Основные функции *разработчика, тестировщика и инженера по качеству*.
3. Стандарт ISO/IEC 12207. Терминология. Классификация и структура процессов ЖЦ ПО.
4. Концепции модульного, структурного и объектно-ориентированного программирования.
5. Что такое CASE ? Приведите классификацию CASE-средств по различным классификационным признакам. Приведите примеры и рассмотрите функциональные возможности известных Вам CASE-средств уровня Low CASE.
6. Определите и дайте краткую характеристику основным *нефункциональным требованиям* к программному продукту.
7. Способы повышения уровня защищенности компьютерных систем от преднамеренных внешних воздействий.
8. Понятие модели жизненного цикла ПО. Особенности каскадной, эволюционной и спиральной моделей.
9. Технология *Rational Unified Process (RUP)*. Иерархическая структура модели ЖЦ RUP: фазы, итерации, рабочие и поддерживающие процессы, практики.
10. Технология *Microsoft Solution Framework (MSF)*. Область применения и особенности. Основные фазы модели ЖЦ MSF.
11. Технология *eXtreme Programming (XP)*. Базовые принципы методов "живой" разработки ПО. Основные фазы модели ЖЦ XP. Правила технологии XP

6.4.2.2 Примеры тестовых заданий для рубежного контроля №1

1. Каков диапазон допустимых значений *однобайтовой* переменной числового типа, представляющей *натуральные* числа ?

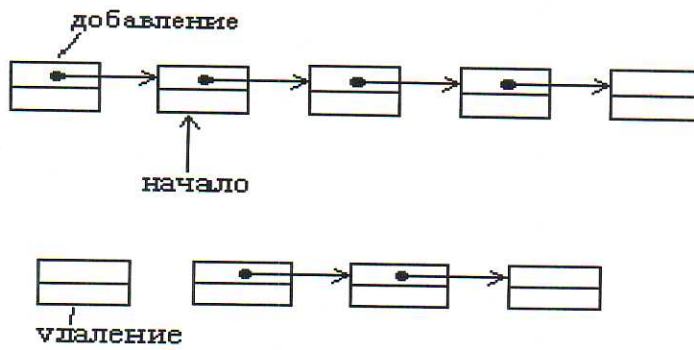
- а) $0 \div 128$
- б) $0 \div 256$
- в) $-127 \div 128$
- г) $0 \div 255$

2. Какая из линейных структур данных представлена на схеме?



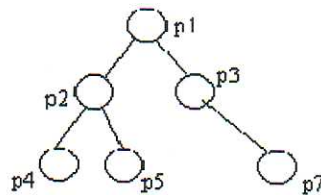
- а) Дек
- б) Очередь
- в) Двусвязный циклический список
- г) Мультисписок

3. Какая из линейных структур данных представлена на схеме?



- а) Дек
- б) Очередь
- в) Стек
- г) Односвязный циклический список

4. Оцените максимальную степень сбалансированности представленного на схеме бинарного дерева,



отнеся его к одной из следующих категорий:

- а) Регулярное дерево поиска
- б) Идеально сбалансированное дерев
- в) Сбалансированное по высоте (АВЛ) дерево
- г) Полное бинарное дерево

5. Сколько операций сравнения потребуется для сортировки по возрастанию элементов следующего массива *методом простого включения* ?

{8;23;5;65}

6. Сколько операций сравнения потребуется для сортировки по возрастанию элементов следующего массива *методом простого обмена* ?

{44;33;1;6}

6.4.2.3 Примеры тестовых заданий для рубежного контроля №2

1. Определите *допустимый диапазон изменения размера файла* данных, обслуживаемого многоуровневым индексом (B⁺-дерево) со следующими параметрами: глубина **H=2**; порядок **P=256**; фактор заполнения $\frac{1}{2} \leq F \leq 1$.

- а) От **1024** до **8192** записей
- б) От **512** до **65 536** записей
- в) От **8192** до **2 097 152** записей
- г) От **4096** до **4 194 304** записей

2. Определите минимально-необходимую *глубину* H B^+ *дерева* для многоуровневого индекса, обслуживающего файл длиной в 4,194,304 (2^{22}) записей, если *порядок индекса* $P=256$, а *фактор заполнения* $F = \frac{1}{2}$ поддерживается на всех уровнях B^+ -дерева, кроме корневого.

- а) $H = 2$
- б) $H = 3$
- в) $H = 4$
- г) $H = 5$

3. Сколько операций сравнения потребуется для поиска всех вхождений подстроки "ax" в строке "0ax123456ax78ax9" *методом прямого поиска?*

6.4.2.4 Примеры тестовых заданий для рубежного контроля №3

3. Какая их типовых моделей жизненного цикла ПО может быть рекомендована к использованию при его разработке в условиях, когда требования заказчика могут изменяться в процессе проектирования ?

- а) Каскадная модель
- б) Эволюционная модель
- в) Спиральная модель
- г) Модель формальной разработки
- д) Ни одна из перечисленных моделей

4. Какая из технологий разработки ПО базируется на следующих принципах:

- *люди, их общение более важны, чем процессы и инструменты;*
- *работающая программа более важна, чем исчерпывающая документация;*
- *сотрудничество с заказчиком более важно, чем обсуждение деталей контракта; отработка изменений более важна, чем следование планам.*

- а) Rational Unified Process (RUP)
- б) Extreme programming (XP)
- в) Microsoft solution framework (MSF)
- г) Ни одна из перечисленных технологий

6.4.2.5 Примеры тестовых заданий для рубежного контроля №4

1. При использовании какого подхода UML предоставляет максимум преимуществ?

- а) Процедурное программирование
- б) Объектно-ориентированное проектирование
- в) Функциональное программирование
- г) Программирование по контракту
- д) Концептуальное проектирование

2. Выберите из списка истинные утверждения, касающиеся UML-моделей:

- а) UML-модели являются XML-документами

- б) UML имеет ограничения по природе моделируемой предметной области
 - в) CASE-средства могут генерировать текстовые спецификации из UML-моделей
 - г) Создавая UML-модель, вы тем самым документируете систему
 - д) UML-модель жестко привязана к конкретной методологии разработки ПО.
3. Что специфицирует *отношение обобщения* между «актерами» на UML-диаграмме вариантов использования?
 4. Что специфицирует *отношение включения* между вариантами использования на UseCase-диаграмме вариантов использования?
 5. Что специфицирует *отношение обобщения* между классами на UML-диаграмме классов?
 6. Определите понятия «событие» и «переход», используемые при разработке State Machine-диаграмм.
 7. Определите понятия «состояние», «составное состояние», «параллельное состояние» и «последовательное состояние», используемые при разработке State Machine-диаграмм.
 8. Определите понятия «историческое состояние», «глубокое историческое состояние», используемые при разработке State Machine-диаграмм.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Основная литература

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. : пер. с англ. – М.: Мир, 1979. – 536 с.
2. Брауде Э. Технология разработки программного обеспечения. пер. с англ. – Санкт-Петербург: Питер, 2004.
3. Вирт Н. Алгоритмы и структуры данных.: пер. с англ. – М.: Мир, 1989. – 360 с.
4. Волк В.К. Практическое введение в программную инженерию: учебное пособие. – СПб.: Изд-во «Лань», 2019. – 100 с.: ил. – (Учебники для вузов. Специальная литература).
5. Давыдов В.Г. Программирование и основы алгоритмизации. Учеб. Пособие. – М.: Высш. шк., 2003. – 447 с.
6. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ / Пер. с англ. под ред. А. Шеня. –М.: МЦНМО, 2002. – 960 с.

7.2. Дополнительная литература

1. Буч Г., Рамбо Дж., Джекобсон А. UML. Руководство пользователя. - ДМК-Пресс, Питер, 2004
2. Буч Г. и др. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 3-е издание: пер. с англ. – Москва: Издательский дом «Вильямс», 2010 – 720 с.
3. Кузнецов С.Д. Методы сортировки и поиска. ИСП РАН, Центр Информационных Технологий. (электронный ресурс)
4. Леоненков А. Самоучитель UML – 2-е издание.: – Санкт-Петербург: БХВ-Петербург.
5. Методы, модели, средства хранения и обработки данных: учебник / Э.Г. Дадян, Ю.А. Зеленков. — М.: Вузовский учебник: ИНФРА-М, 2017. – 168 с.
6. Соммервилл И. Инженерия программного обеспечения – 6-е издание: пер. с англ. – Москва: Издательский дом «Вильямс», 2002. – 624 с.: ил

7.3 Методические материалы

1. Семахин А.М. Структуры и алгоритмы обработки данных: Методические указания к выполнению лабораторных и контрольных работ для студентов специальности 230105. КГУ, 2010 г.

7.4 Информационно-справочные материалы

1. ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств.
2. StarUML™. The Open Source UML/MDA Platform. Руководство пользователя (электронный ресурс).
3. UML 2.5 Diagrams Overview. URL: <https://www.uml-diagrams.org/>.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение дисциплины включает в себя учебные лаборатории и классы, оснащенные современными компьютерами (все – в стандартной комплектации для лабораторных занятий и самостоятельной работы), объединенными в локальную вычислительную сеть с выходом в Интернет; мультимедийное оборудование (переносной персональный компьютер, мультимедийный проектор, проекционный экран).

Программные средства обеспечения учебного процесса включают: операционные системы (Windows); инструментальные средства программирования и CASE-средства проектирования (StarUMLTM. The Open Source UML/MDA Platform); вспомогательные (программы презентационной графики; текстовые редакторы; графические редакторы).

Аннотация
рабочей программы учебной дисциплины
ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

образовательной программы высшего образования –
программы специалитета 10.05.03

Информационная безопасность автоматизированных систем

Направленность:

*Обеспечение информационной безопасности
распределенных информационных систем*

Форма обучения: **очная**

Трудоемкость дисциплины: 6 ЗЕ (216 академических часа)

Семестры: 3-й и 4-й

Форма промежуточной аттестации: Экзамен (3-й семестр), Экзамен и защита курсовой работы (4-й семестр),

Содержание дисциплины

Раздел 1 Типы и структуры данных в высокоуровневых языках программирования

Введение. Понятие типа данных в высокоуровневых языках программирования. Линейные структуры данных. Иерархические структуры данных (деревья). Сетевые структуры данных (графы).

Раздел 2 Алгоритмы сортировки данных

Методы внутренней сортировки. Методы внешней сортировки.

Раздел 3 Алгоритмы поиска данных

Алгоритмы поиска подстроки в строке. Методы поиска на основе деревьев
Алгоритмы поиска на графах

Раздел 4 Технологии промышленной разработки ПО

Предмет и основные понятия программной инженерии. Жизненный цикл ПО: базовые понятия. Модели жизненного цикла ПО. Визуальное моделирование при анализе и проектировании программных систем. Выявление и анализ функциональных требований. UML-диаграммы вариантов использования. Разработка логической модели программной системы. UML-диаграммы классов. Разработка динамических моделей программной системы. Разработка моделей физического представления программной системы.