

Махачкала 2023

ЛИСТ РАССМОТРЕНИЯ И СОГЛАСОВАНИЯ

Рабочая программа составлена на основании требований Федерального государственного образовательного стандарта к содержанию и уровню подготовки выпускников по направлению подготовки 09.03.03 «Прикладная информатика» утвержденного приказом Министерства образования и науки РФ №207 от 12.03.2015 года и с учетом зональных особенностей Республики Дагестан.

Составитель:

канд. экон. наук, доцент



Бачиев Б.А.

Рабочая программа обсуждена на заседании кафедры информатики и цифровых технологий № 7 от 12.03. 2023 г.

Заведующий кафедрой:



Н.А. Юсуфов

Рабочая программа одобрена методической комиссией экономического факультета протокол №7 от 15.03.2023г.

Председатель методической комиссии
факультета, Азракулиев З. М.



Содержание

1. Цели и задачи дисциплины	4
2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы	4
3. Место дисциплины в структуре образовательной программы	6
4. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся	6
5. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий	7
5.1. Разделы дисциплины по видам занятий	7
5.2. Тематический план лекций	7
5.3. Тематический план практических (лабораторных, семинарских) занятий	9
5.4. Содержание разделов дисциплины	10
6. Учебно-методическое обеспечение самостоятельной работы	10
7. Фонды оценочных средств	12
7.1. Описание показателей и критериев оценивания компетенций	12
7.2. Типовые контрольные задания (тесты, коллоквиумы и т.д.)	14
7.3. Методика оценивания знаний, умений, навыков	20
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	22
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	23
10. Методические указания для обучающихся по освоению дисциплины	24
11. Информационные технологии и программное обеспечение	25
12. Описание материально-технической базы необходимой для осуществления образовательного процесса по дисциплине	25
13. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья	25
Дополнения и изменения в рабочую программу дисциплины	27

1. Цели и задачи дисциплины

Целью курса «Языки программирования» является подготовка специалиста к деятельности, связанной созданием приложений в различных средах программирования. Она является базовой для изучения дисциплин «Методы программирования», программно-аппаратные методы защиты информации, моделирование систем безопасности, операционные системы и системы управления базами данных. Знания и практические навыки, полученные из курса «Языки программирования», используются обучаемыми при изучении естественнонаучных дисциплин, а также при разработке курсовых и дипломных работ.

Задачи дисциплины «Языки программирования» следующие:

- знакомство с методами структурного и объектно-ориентированного программирования, как наиболее распространенными и эффективными методами разработки программных продуктов;
- закрепление навыков проектирование и программирования на основе языка программирования высокого уровня;
- знакомство с основными структурами данных и типовыми методами обработки этих структур;
- обучение разработке алгоритмов на основе объектно-ориентированного подхода;

2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины студент должен обладать следующим набором компетенций:

- способностью к самоорганизации и самообразованию (ПК 2);
- способностью применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий (ПК-2);
- способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям (ПК 2);
- способностью применять на практике международные и профессиональные стандарты информационных технологий, современные

парадигмы и методологии, инструментальные и вычислительные средства (ПК 2).

В результате указанных компетенций обучающийся должен:

Компетенции	Содержание компетенции (или ее части)	Раздел дисциплины, обеспечивающий этапы формирования компетенции	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции (или ее части) обучающийся должен:		
			знать	уметь	владеть
ПК 2	ИД 1 ПК 2 Знает структуры и методы компьютерной обработки данных, внедрения ИС, различные языки программирования, принципы и методы разработки системного и прикладного программного обеспечения		Синтаксис выбранного языка программирования, особенности программирования на этом языке; методологии разработки программного обеспечения и технологии программирования; различные парадигмы программирования; системы классификации и кодирования данных	Предсказывать результат выполнения незнакомых языковых конструкций; осваивать новые языки программирования	Чтения и написания программного кода с применением различных парадигм программирования; использования особенностей языка программирования для построения эффективных программ
	ИД 2 ПК 2 Умет внедрять ИС, достигать стратегические цели, разрабатывать алгоритмы и программные решения;			Читать программные коды; применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования	Чтения, понимания и верификации данного программного кода; использования особенностей языка программирования для построения эффективных программ

ИД 3 ПК 2 Владеет навыками работы в различных программных средах, разработки алгоритмов и программ, отладки и тестирования компьютерных программ.		Стандартные библиотеки выбранного языка программирования; особенности различных языков программирования и их влияние на эффективность разрабатываемого программного кода	Применять выбранные языки программирования для реализации заданных алгоритмов	Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов
--	--	--	---	--

3. Место дисциплины в структуре образовательной программы

Дисциплина относится к дисциплинам профессионального цикла. Изучение дисциплины требует входных компетенций, знаний, умений и навыков, предусмотренных школьным курсом информатики. Сама же она необходима для следующих дисциплин учебного плана:

№ п/п	Наименование обеспечиваемых (последующих) дисциплин	№ № разделов данной дисциплины, необходимых для изучения (последующих) обеспечиваемых дисциплин	
		1	2
1.	Интернет технологии		
2.	Веб-программирование		
3.	Методы и средства проектирования информационных систем и технологий		
4.	Моделирование и исследования с использованием прикладных программ		

4. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Очная форма обучения

Вид учебной работы	Всего часов	Семестр 1
Общая трудоемкость: часы	180	
зачетные единицы	5	
Аудиторные занятия (всего), в т.ч.:	64	

Лекции	32	
Практические занятия (ПЗ)	32	
Лабораторные занятия (ЛР)		
Семинарские (С)		
Самостоятельная работа (СРС), в т.ч.:**	116	
подготовка к практическим занятиям	50	
самостоятельное изучение тем	66	
курсовая работа (проект)		
подготовка к текущему контролю	14	
Промежуточная аттестация		Зачет с оценкой

5. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

5.1. Разделы дисциплины по видам занятий

Очная форма обучения

№ п/п	Наименование разделов	Всего (часов)	Аудиторные занятия (час)		Само- стоя- тель- ная ра- бота
			Лекции	ЛПЗ	
1.	Основы C++		10	10	
2.	ООП в C++		10	10	
3.	Библиотеки в C++		12	12	
	Всего		32	32	

5.2. Тематический план лекций

Очная форма обучения

п/п	Темы лекций	Количество часов
Основы C++		
1.	Введение в язык программирования C++ История создания языка и его эволюция. Международный стандарт языка. Сферы применения языка Си++. Пример простой программы. Объясняется процесс ее написания, использования простейших конструкций языка, использования транслятора и запуск программы на выполнение. Правила именования переменных и функций языка, правила записи констант. Понятие ключевого или зарезервированного слова, список ключевых слов Си++.	2
2.	Операции и выражения. Встроенные типы данных. Правила формирования и вычисления выражений в языке Си++. Все операции языка. Рассматриваются все встроенные типы языка	4

	Си++: целые числа разной разрядности, вещественные числа, логические величины, перечисляемые значения, символы и их кодировка. Операция условия: ?::.	
3.	Ввод-вывод. Потоки. Манипуляторы и форматирование ввода-вывода. Строковые потоки. Ввод-вывод файлов. Ввод и вывод одного символа. Буферы. Чтение одной строки. Чтение файла. Переключение и работа с файлами.	2
4.	Операторы языка С++. Операторы управления, имеющиеся в языке Си++.	4
5.	Операторы языка С++. Операторы построения циклов, имеющиеся в языке Си++.	2
6.	Функции. Функции – это основные единицы построения программ при процедурном программировании на языке Си++. Правила их записи, вызова и передачи параметров.	2
7.	Распределение памяти. Работа с областями памяти и строками. Проблемы при явном распределении памяти в Си++, способы их решения. Ссылки и указатели. Операции с указателями. Распределение памяти под переменные, управление памятью с помощью переопределения операторов new и delete. Создание и использование массивов, структур, объединений, указателей. Адресная арифметика. Строки и литералы. Строки - дополнительные сведения о связи между указателями и массивами. Инициализация массивов и классы памяти.	4
8.	Стандартная библиотека С++. Описание стандартных функций.	2
9.	Пространство имен. Возможности построения больших программ, использование общих данных. Определение контекста, оператор namespace.	2
ООП в С++		
10.	Классы, объекты и структуры. Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов. Наследование, виды наследования. Виртуальные методы. Абстрактные классы. Множественное наследование	2
11.	Дополнительные возможности классов. Конструкторы и деструкторы классов. Возможности инициализации объектов. Копирующий конструктор. Операции new и delete. Дополнительные возможности при определении классов, включая переопределение операций, определение методов inline и задание собственных преобразований типа	2
12.	Перегрузка операций. Лекция содержит описание механизма перегрузки операций в С++. Программист может задать интерпретацию операций, когда они применяются к объектам определенного класса. Помимо арифметических, логических и операций отношения можно переопределить вызов функций (), индексацию [], косвенное обращение ->, а также присваивание и инициализацию. Можно определить явные и скрытые преобразования между пользовательскими и основными типами. Показано, как определить класс, объект которого можно копировать и уничтожать только с помощью специальных, определенных пользователем функций.	2

13.	Пример реализации библиотеки для работы с комплексными числами. Создание класса для комплексного числа.	2
Библиотеки в C++		
14.	Препроцессор. Общие сведения. Символические константы: #define. Использование аргументов с #define. Макроопределение или функция? Включение файла #include. Условная компиляция.	2
15.	Обработка ошибок. Попытка классификации ошибок. Сообщение об ошибке с помощью возвращаемого значения. Исключительные ситуации. Обработка исключительных ситуаций, операторы try и catch.	2
16.	Шаблоны. Понятие шаблона. Функции-шаблоны. Шаблоны классов.	2
17.	Проектирование библиотек. Эта лекция содержит описание различных приемов, оказавшихся полезными при создании библиотек для языка C++. В частности, в ней рассматриваются конкретные типы, абстрактные типы, узловые классы, управляющие классы и интерфейсные классы. Помимо этого обсуждаются понятия обширного интерфейса и структуры области приложения, использование динамической информации о типах и методы управления памятью. Внимание акцентируется на том, какими свойствами должны обладать библиотечные классы, а не на специфике языковых средств, которые используются для реализации таких классов и не на определенных полезных функциях, которые должна предоставлять библиотека.	2
18.	Понятие о библиотеках STL и BOOST. Описание принципов использования STL и BOOST	2
Всего часов		

5.3. Тематический план практических (лабораторных, семинарских) занятий

Очная форма обучения

п/п	Темы практических (лабораторных, семинарских) занятий	Количество часов
Основы C++		
1.	Ознакомление со средой программирования	2
2.	Форматный и файловый ввод-вывод	4
3.	Работа с операторами ветвления	4
ООП в C++		
4.	Работа с циклами	5
5.	Работа с динамическими массивами	4
6.	Работа с классами	4
Библиотеки в C++		
7.	Работа с пространством имен	4
8.	Проектирование библиотеки	4
Всего часов		32

5.4. Содержание разделов дисциплины

№ п/п	Наименование раздела (темы)	Содержание раздела	Компетенции
1.	Основы С++	Введение в язык программирования С++ История создания языка и его эволюция. Международный стандарт языка. Сферы применения языка Си++. Пример простой программы. Объясняется процесс ее написания, использования простейших конструкций языка, использования транслятора и запуск программы на выполнение. Правила именования переменных и функций языка, правила записи констант. Понятие ключевого или зарезервированного слова, список ключевых слов Си++.	ПК 1 ИД 1 ПК 2 ИД 2 ПК 2 ИД 3 ПК 2
2.	ООП в С++	Классы, объекты и структуры. Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов. Наследование, виды наследования. Виртуальные методы. Абстрактные классы. Множественное наследование.	ПК 1 ИД 1 ПК 2 ИД 2 ПК 2 ИД 3 ПК 2
3.	Библиотеки в С++	Понятие о библиотеках STL и BOOST. Описание принципов использования STL и BOOST	ПК 1 ИД 1 ПК 2 ИД 2 ПК 2 ИД 3 ПК 2

6. Учебно-методическое обеспечение самостоятельной работы

Тематический план самостоятельной работы

п/п	Тематика самостоятельной работы	Количество часов	Рекомендуемые источники информации (№ источника)		
			основная (из п.8 РПД)	дополнительная (из п.8 РПД)	(интернет-ресурсы) (из п.9 РПД)
1	Препроцессорные средств	12	1,2,3	4,5,6	1-7
2	Структуры и объединения	12			
3	Форматированный ввод-вывод в С++.	12			
4	Обработка исключительных ситуаций.	12			
5	Графическая библиотека С++.	10			
6	Локальные классы	12			
7	Классы и шаблоны.	12			
8	Форматирование данных при обмене потоков.	12			

9	Конструкторы и деструкторы в включениях.	12			
10	Динамическая идентификация типов.	10			
	Всего	116			

Учебно-методические материалы для самостоятельной работы:

Самостоятельная работа студентов, предусмотренная учебным планом в объеме не менее 50-70% общего количества часов, соответствует более глубокому усвоению изучаемого курса, формирует навыки исследовательской работы и ориентирует студентов на умение применять теоретические знания на практике.

Самостоятельная работа носит систематический характер.

Результаты самостоятельной работы контролируются преподавателем и учитываются при аттестации студента (зачет, экзамен). При этом проводятся: тестирование, экспресс-опрос на семинарских и практических занятиях, заслушивание докладов, рефератов, проверка письменных работ и т.д.

Задания для самостоятельной работы составляются по разделам и темам, по которым не предусмотрены аудиторские занятия, либо требуется дополнительно проработать и проанализировать рассматриваемый преподавателем материал в объеме запланированных часов.

Для подготовки к занятиям и выполнения самостоятельной работы, студентам рекомендуются учебно-методические издания, а также методические материалы, выпущенные кафедрой своими силами и предоставляемые студентам во время занятий (приложения):

- наглядные пособия (плакаты, гербарий - на кафедре)
- глоссарий - словарь терминов по тематике дисциплины
- тезисы лекций.

Самостоятельная работа с книгой. В наше время книга существует в двух формах: традиционной и электронной. В интернете существуют целые библиотеки, располагающие десятками тысяч электронных текстов. Сегодня в обществе преобладает мнение, что печатная книга и ее компьютерный текст дополняют друг друга. Используя электронный вариант книги значительно быстрее подготовить на его базе реферат, контрольную работу, подогнать текст своей работы под требуемый учебным заданием объем. Печатные книги гораздо легче и удобнее читать.

Работа с книгой, студенты сталкиваются с рядом проблем. Одна из них – какая книга лучше. Целесообразно в первую очередь обратиться к литературе, рекомендованной преподавателем. Целесообразно прочитать аннотацию к книге на ее страницах, в которой указано, кому и для каких целей она может быть полезна.

Другая проблема – как эффективно усвоить материал книги. Качество усвоения учебного материала существенно зависят от манера прочтения книги. Можно выделить пять основных приемов работы с литературой:

Чтение-просмотр используется для предварительного ознакомления с книгой, оценки ее ценности. Он предполагает ознакомление с аннотацией, предисловием, оглавлением, заключением книги, поиск по оглавлению наиболее важных мыслей и выводов автора произведения.

Выборочное чтение предполагает избирательное чтение отдельных разделов текста. Этот метод используется, как правило, после предварительного просмотра книги, при ее вторичном чтении.

Сканирование представляет быстрый просмотр книги с целью поиска фамилии, факта, оценки и др.

Углубленное чтение предполагает обращение внимания на детали содержания текста, его анализ и оценку. Скорость подобного вида чтения составляет ориентировочно до 7-10 страниц в час. Она может быть и выше, если читатель уже обладает определенным знанием по теме книги или статьи.

Углубленное чтение литературы предполагает:

- Стремление к пониманию прочитанного. Без понимания смысла прочитанного информацию ее очень трудно запомнить.

- Обдумывание изложенной в книге информации. Тогда собственные мысли, возникшие в ходе знакомства с чужими работами, послужат основой для получения нового знания.

- Мысленное выделение ключевых слов, идей раздробление содержания текста на логические блоки, составление плана прочитанного. Если студент имеет дело с личной книгой, то ключевые слова и мысли можно подчеркнуть карандашом.

- Составление конспекта изученного материала. Если статья или раздел книги по объему небольшой, то целесообразно приступить к конспектированию, прочитав их полностью. В других случаях желательно прочитать 7-10 страниц.

7. Фонды оценочных средств

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенции по завершении освоения дисциплины;
- эталонный уровень характеризуется максимально возможной выраженностью компетенции и является важным качественным ориентиром для самосовершенствования.

Уровень сформированности каждой компетенции на различных этапах ее формирования в процессе освоения данной дисциплины оценивается в ходе текущего контроля успеваемости и представлен различными видами оценочных средств.

7.1. Описание показателей и критериев оценивания компетенций

Показатели	Критерии оценивания			
	Шкала по традиционной пятибальной системе			
	допороговый («неудовлетворительно»)	Пороговый («удовлетворительно»)	Продвинутый («хорошо»)	Высокий («отлично»)
ПК 1				
Знания	Фрагментарные знания по современным языкам программирования	Общие сведения о лексике, синтаксисе и семантике языка программирования	Общие сведения о лексике, синтаксисе и семантике языка программирования; парадигмы программирования	Общие сведения о лексике, синтаксисе и семантике языка программирования; парадигмы программирования; варианты типизации, исполнения конструкций, диспетчеризации в языках программирования
Умения	Отсутствие умений, предусмотренных данной компетенцией	Предсказывать результат выполнения незнакомых языковых конструкций;	Предсказывать результат выполнения незнакомых языковых конструкций;	Предсказывать результат выполнения незнакомых языковых конструкций; осваивать новые языки программирования
Навыки	Отсутствие или наличие фрагментарных навыков предусмотренных данной компетенцией	Чтения программного кода с применением различных парадигм программирования;	Чтения и написания программного кода с применением различных парадигм программирования;	Чтения и написания программного кода с применением различных парадигм программирования; использования особенностей языка программирования для построения эффективных программ
ПК-2				
Знания	Фрагментарные знания по современным языкам программирования	Синтаксис выбранного языка программирования, особенности программирования на этом языке	Синтаксис выбранного языка программирования, особенности программирования на	Синтаксис выбранного языка программирования, особенности программирования на

			этом языке; методологии разработки программного обеспечения и технологии программирования	этом языке; методологии разработки программного обеспечения и технологии программирования; различные парадигмы программирования; системы классификации и кодирования данных
Умения	Отсутствие умений, предусмотренных данной компетенцией	Читать программные коды	Читать программные коды; применять выбранные языки программирования для написания программного кода	Читать программные коды; применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования
Навыки	Отсутствие или наличие фрагментарных навыков предусмотренных данной компетенцией	Чтения, понимания и данного программного кода	Чтения, понимания и верификации данного программного кода	Чтения, понимания и верификации данного программного кода; использования особенностей языка программирования для построения эффективных программ

7.2. Типовые контрольные задания (тесты, коллоквиумы и т.д.)

Контрольные работы:

Темы:

1. Основные концепции языков программирования.

Опрос на лекции:

Темы:

1. Классификация языков программирования.
2. Средства описания данных и действий.

3. Основные концепции объектно-ориентированного программирования. Абстрактные типы данных.
4. Обработка файлов.
5. Обработка исключительных ситуаций.
6. Параллельное программирование.
7. Макропроцессоры и макрогенераторы.
8. Современные интегрированные среды разработки программ. Графический интерфейс пользователя.
9. Отладчики; генераторы кода/приложений.
10. Библиотеки программ и классов.
11. Язык Ассемблер.
12. Основы логического программирования.
13. Основы функционального программирования.
14. Стандарты языков программирования и документирование программ.

Опросы на практическом занятии:

Темы:

1. Анализ задачи. Абстракция программ и данных. Синтаксис языка программирования.
2. Обработка текстов на естественном языке, синтаксические анализаторы.
3. Вещественные числа. Ошибки при работе с вещественными числами.
4. Рекурсия. Типы рекурсий.
5. Генерирование и обработка исключительных ситуаций.
6. Объектно-ориентированное программирование. Модель предметной области и модель проектирования.
7. Основные приемы объектно-ориентированного программирования.
8. Язык логического программирования Prolog.
9. Язык функционального программирования Haskell.

Отчет по индивидуальной практической работе:

Темы:

1. Основные концепции, история, сайты, компиляторы и расширения языка «Фортран».
2. Основные концепции, история, сайты, компиляторы и расширения языка «Пролог».
3. Основные концепции, история, сайты, компиляторы и расширения языка «Java».
4. Основные концепции, история, сайты, компиляторы и расширения языка «ASP».
5. Основные концепции, история, сайты, компиляторы и расширения языка «PHP».
6. Основные концепции, история, сайты, компиляторы и расширения

- языка «Perl».
7. Основные концепции, история, сайты, компиляторы и расширения языка «Ruby».
 8. Основные концепции, история, сайты, компиляторы и расширения языка «Python».
 9. Построение компиляторов. Основные концепции.
 10. Компиляторы для C++. Сравнительный анализ.
 11. Языки программирования для сети Интернет. Сравнительный анализ.
 12. Объектно-ориентированное программирование. Основные положения, правила создания классов и объектов.
 13. Защита программ от несанкционированного копирования.
 14. Защита программ от анализа.
 15. Взлом программ с защитами. Дизассемблеры.
 16. Оценка качества программ. Методы, стандарты и основные положения.
 17. Методы оценки стоимости программных продуктов.
 18. Оценка сложности вычисления операций.
 19. Методы оценки трудоемкости создания программ.
 20. Нормативы написания исходного кода.
 21. Документирование программ. Основные стандарты.
 22. Распространение программ. Shareware, Freeware.
 23. Вычисление контрольных сумм файлов программ. Алгоритмы и реализации.
 24. Шуточные и эзотерические языки программирования.
 25. Методы обфускации исходного кода.
 26. Программирование для операционной системы MacOS.
 27. Программирование для мобильной платформы Android.
 28. Облачные вычисления.
 29. Стандарт HTML5.
 30. Программирование искусственного интеллекта.
 31. Унифицированный язык моделирования UML.
 32. Модель компонентных объектов COM.
 33. Интеллектуальный анализ данных (Data Mining).
 34. Сравнительный анализ языков функционального программирования.
 35. Обработка текстов на естественных языках.
 36. Основные концепции, история, сайты, компиляторы и расширения языка «Scala»
 37. MPI (Message Passing Interface)
 38. Автоматическое построение онтологий для извлечения знаний из текста.
 39. Библиотека Qt.
 40. Параллельное программирование для графических ускорителей. Технология CUDA.
 41. Кроссплатформенное программирование. Сравнительный анализ

- языков программирования.
42. Сравнительный анализ языков логического программирования.
 43. Искусственные нейронные сети. Современное состояние вопроса.
 44. Динамические библиотеки.
 45. Технология ActiveX.
 46. Методологии тестирования программного обеспечения.
 47. Автоматизированное тестирование программного обеспечения.
 48. Основные концепции, история, сайты, компиляторы и расширения языка «R».
 49. Основные подходы и принципы разработки дизайна программного обеспечения
 50. Особенности, основные подходы и принципы разработка видеоигр.
 51. Системы контроля версий программ. Сравнительный анализ.
 52. Continuous Integration, непрерывная интеграция и автоматизация сборок проекта.
 53. Методы монетизации программного обеспечения.
 54. Программирование для мобильной платформы iOS.
 55. Программирование для мобильной платформы Windows Phone.
 56. Основные концепции, история, сайты, компиляторы и расширения языка «Wolfram language».
 57. Аудит информационной безопасности веб-сайта.
 58. Паттерны объектно-ориентированного проектирования.
 59. Программирование микроконтроллеров.
 60. Управление разработкой программного обеспечения. Стандарты, методы, особенности.

Экзамен:

Вопросы к экзамену:

1. Язык программирования. Общие принципы построения и использования языков программирования.
2. Стандарты языков программирования.
3. Лямбда-исчисление. Аппликация, абстракция, редукция, преобразование.
4. Списки и функциональные выражения в функциональных языках программирования.
5. Механизмы и средства взаимодействия программы с операционной системой.
6. Функциональное программирование. Основные положения. Основные отличия от других типов языков программирования.
7. Классификация языков программирования. Близость языков программирования к естественному языку.
8. Унификация и хорновский клоз в логических языках программирования.
9. Модель вычислений функциональных языков программирования.
10. Языки программирования низкого уровня.

11. Средства разработки графического интерфейса пользователя. Эргономические свойства человеко-машинного интерфейса.
12. Процедурные языки программирования. Основные отличия от других типов языков.
13. Обоснование выбора языка программирования.
14. Перегрузка в языках программирования.
15. Логические языки программирования. Основные положения и понятия. Отличия от других типов языков программирования.
16. Объектно-ориентированные языки программирования. Основные отличия от других концепций языков программирования.
17. Полиморфизм в объектно-ориентированном программировании. Виртуальные функции. Таблицы виртуальных функций.
18. Структура языка программирования. Синтаксис и семантика языков программирования. Расширенная форма Бэкуса-Наура.
19. Наследование в объектно-ориентированном программировании. Множественное наследование. Проблемы множественного наследования.
20. Инкапсуляция в объектно-ориентированном программировании. Контроль доступа.
21. Понятие класса и объекта в объектно-ориентированном программировании. Атрибуты, методы, конструктор и деструктор, статические члены класса.
22. Диаграммы классов UML. Основные элементы и обозначения.
23. Данные. Средства описания данных. Типизация языка.
24. Преобразование типов. Контроль соответствия типов данных.
25. Объектно-ориентированное программирование. Основные концепции объектно-ориентированного программирования.
26. Современные интегрированные среды разработки программ. Основные компоненты среды программирования.
27. Декомпозиция программ.
28. Трансляторы. Интерпретация и компиляция.
29. Макропроцессоры и макрогенераторы.
30. Потоки и процессы. Сходства и различия.
31. Мониторы и защищаемые переменные в параллельном программировании.
32. Семафоры в параллельном программировании. Типы семафоров.
33. Отладчики. Генераторы кода и приложений.
34. Параллельная обработка данных и параллелизм. Параллельное и распределенное программирование.
35. Основные проблемы параллельного и распределенного программирования.
36. Оценка максимально возможного параллелизма.
37. Основные модели параллельного программирования.
38. Оптимизатор. Основные функции оптимизатора.
39. Обработка исключительных ситуаций. Иерархия исключительных ситуаций. Виды исключительных ситуаций.

40. Операторы обработки исключительных ситуаций в различных языках программирования.
41. Схема обработки исключительных ситуаций Б. Мейера.
42. Элементарные типы данных.
43. Перегрузка данных, операторов, методов.
44. Составные типы данных.
45. Механизмы логического вывода. Прямая и обратная цепочки рассуждений.
46. Пространство имен, область видимости, время жизни переменных.
47. Ошибки при работе с вещественными числами. Смешанная арифметика.
48. Операторы выбора и условные операторы.
49. Вещественные числа. Способы представления. Операции над вещественными числами.
50. Оператор присваивания. Операторы цикла.
51. Распределение памяти при выполнении программы.
52. Куча. Менеджер кучи. Фрагментация динамической памяти.
53. Концепция виртуальной памяти. Страничная организация памяти.
54. Сегментный принцип организации памяти. Сегментация памяти.
55. Указатели. Операции над указателями. Типизированные и нетипизированные указатели.
56. Динамические структуры данных. Реализация динамических структур данных с помощью указателей.
57. Библиотеки программ и классов. Статические и динамические библиотеки. Критерии проектирования библиотек.
58. Подпрограммы. Формальные и фактические параметры подпрограмм.
59. Передача параметров подпрограмме.
60. Программный стек и его изменение.
61. Рекурсивный и итерационный методы решения задач. Виды рекурсий.
62. Общая характеристика языков ассемблера: назначение, принципы построения и использования; структура языка.
63. Сериализация и десериализация. Методы сериализации объектов в базу данных.
64. Динамическая диспетчеризация.
65. Родовые (настраиваемые) сегменты.
66. Шаблоны.
67. Вариантные записи.
68. Средства описания действий над данными. Операторы, выражения, модули, блоки в языках программирования.
69. Ленивые и жадные вычисления в процедурном и функциональном программировании.
70. Языки высокого уровня.
71. Показатели качества программных средств.

72. Отношения между классами в объектно-ориентированном программировании.
73. Файлы, обработка файлов. Типы доступа к файлам.
74. Абстрактные типы данных: инкапсуляция, спецификация, реализация, параметризация.
75. Основные группы команд, операторы, средства взаимодействия с операционной системой в языках ассемблера.

7.3. Методика оценивания знаний, умений, навыков

Оценка знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций по дисциплине проводятся в форме текущего контроля и промежуточной аттестации. Текущий контроль проводится в течение семестра с целью определения уровня усвоения обучающимися знаний, формирования умений и навыков, своевременного выявления преподавателем недостатков в подготовке обучающихся и принятия необходимых мер по ее корректировке, а также для совершенствования методики обучения, организации учебной работы и оказания индивидуальной помощи обучающимся.

Критерии оценки знаний студентов при проведении тестирования

Оценка «отлично» выставляется при условии правильного ответа студента не менее чем 85% тестовых заданий.

Оценка «хорошо» выставляется при условии правильного ответа студента не менее чем 70% тестовых заданий.

Оценка «удовлетворительно» выставляется при условии правильного ответа студента не менее чем 50% тестовых заданий.

Оценка «неудовлетворительно» выставляется при условии правильного ответа студента менее чем 50% тестовых заданий.

Критерии оценки знаний студента при написании индивидуального задания

Оценка «отлично» - выставляется студенту, показавшему всесторонние систематизированные, глубокие знания вопросов и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «хорошо» - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике. Но допускает в ответе или в решении задач некоторые неточности, которые может устранить с помощью дополнительных вопросов преподавателя.

Оценка «удовлетворительно» - выставляется студенту. Показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала. Но при этом он владеет основными понятиями выносимых на контрольную работу тем, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации.

Оценка «неудовлетворительно» - выставляется студенту, который не знает большей части основного содержания выносимых на контрольную работу вопросов тем.

Критерии оценки ответов на экзамене

Оценка **«отлично»** выставляется студенту, который:

1) глубоко, в полном объеме освоил программный материал, излагает его на высоком научно-теоретическом уровне, изучил обязательную и дополнительную литературу, умеет правильно использовать знания при применение информационных технологий в экономике, ориентируется в современных проблемах практического применения персональных компьютеров и информационных технологий;

2) умело применяет теоретические знания по информатики при решении практических задач в экономике;

3) владеет современными методами применения ПК и информационных технологий в экономике, самостоятельно пополняет и обновляет знания в ходе учебной работы;

4) при освещении второстепенных вопросов возможны одна две неточности, которые студент легко исправляет после замечания преподавателя.

Оценку **«хорошо»** получает студент, который:

1) раскрыл содержание вопроса в объеме, предусмотренном программой, изучил обязательную литературу по информатике;

2) грамотно изложил материал, владеет терминологией;

3) знаком с методами применения ПК в прикладных задачах экономики, умеет увязать теорию с практикой;

4) в изложении допустил ряд неточностей, не искажающих содержания ответа на вопрос.

Оценка **«удовлетворительно»** ставится студенту, который:

1) освоил программный материал по информатике в объеме учебника, обладает достаточными для продолжения обучения и предстоящей профессиональной деятельности знаниями, выполнил текущие задания;

2) при ответе допустил несущественные ошибки, неточности, нарушения последовательности изложения материала, недостаточно аргументировано изложил теоретические положения.

Оценка **«неудовлетворительно»** выставляется студенту, который:

- 1) обнаружил значительные пробелы в знании основного программного материала;
- 2) допустил принципиальные ошибки в выполнении предусмотренных программой заданий.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) Основная литература:

1. Бессмертный, И. А. Системы искусственного интеллекта : учеб. пособие для СПО / И. А. Бессмертный. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2018. — 130 с.
2. Гниденко, И. Г. Технология разработки программного обеспечения : учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 235 с.
3. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2 : учебник для вузов / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2019. — 501 с.
4. Жмудь, В. А. Моделирование замкнутых систем автоматического управления : учеб. пособие для академического бакалавриата / В. А. Жмудь. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2019. — 128 с.
5. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для академического бакалавриата / С. В. Зыков. — М. : Издательство Юрайт, 2019. — 155 с.

б) Дополнительная литература:

1. Иванов, В. М. Интеллектуальные системы : учеб. пособие для СПО / В. М. Иванов ; под науч. ред. А. Н. Сесекина. — М. : Издательство Юрайт, 2019. — 93 с.
2. Иванов, В. М. Интеллектуальные системы : учеб. пособие для вузов / В. М. Иванов ; под науч. ред. А. Н. Сесекина. — М. : Издательство Юрайт, 2017. — 91 с.

3. Кубенский, А. А. Функциональное программирование : учебник и практикум для академического бакалавриата / А. А. Кубенский. — М. : Издательство Юрайт, 2019. — 348 с.

4. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учеб. пособие для СПО / Е. В. Кудрина, М. В. Огнева. — М. : Издательство Юрайт, 2019. — 322 с.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Elibrary. ru (РИНЦ)- научная электронная библиотека. – Москва, 2000. <http://elibrary.ru>
2. Мировая цифровая библиотека - <https://www.wdl.org/ru/country/RU/>
3. Научная библиотека МГУ имени М.В. Ломоносова - <http://nbmgu.ru/>
4. Российская государственная библиотека - rsl.ru
5. Бесплатная электронная библиотека - Единое окно доступа к образовательным ресурсам - <http://window.edu.ru/>

	Наименование электронно-библиотечной системы (ЭБС)	Принадлежность	Адрес сайта	Наименование организации-владельца, реквизиты договора на использование
1	Электронно-библиотечная система «Издательство Лань» («Ветеринария и сельское хозяйство»)	сторонняя	http://e.lanbook.com	ООО «Издательство Лань» Санкт-Петербург Договор № 112/140/2017, от 25/10/2017 21.12.2017 по 20.12.2018гг
2	Электронно-библиотечная система «Издательство Лань» («Лесное хозяйство и лесоинженерное дело»)	сторонняя	http://e.lanbook.com	ООО «Издательство Лань» Санкт-Петербург Соглашение № 21 от 21.12.2017г 21.12.2017 по 20.12.2018гг
3	Электронно-библиотечная система «Издательство Лань». «Технология пищевых производств».	сторонняя	http://e.lanbook.com	ООО «Издательство Лань» Санкт-Петербург Договор № 46 от 20/04/2018 с 15/05/18 до 14/05/19

10. Методические указания для обучающихся по освоению дисциплины

Дисциплина предусматривает лекции каждую неделю и лабораторные работы раз в две недели, а также выполнение расчетно-графической работы. Изучение курса завершается зачетом.

Успешное изучение курса требует посещения лекций, активной работы на лабораторных работах, выполнения всех учебных заданий преподавателя, ознакомления с основной и дополнительной литературой.

Во время лекции студент должен вести краткий конспект. Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции.

Обучающемуся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Лабораторные работы составляют важную часть профессиональной подготовки студентов. Они направлены на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений.

Выполнение студентами лабораторных работ направлено на:

- обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам дисциплин;
- формирование необходимых профессиональных умений и навыков;

Содержание лабораторных работ фиксируется в разделе 4 настоящей рабочей программы.

Необходимыми структурными элементами лабораторной работы, помимо самостоятельной деятельности студентов, являются инструктаж, проводимый преподавателем, а также организация обсуждения итогов выполнения лабораторной работы.

Выполнению лабораторных работ предшествует проверка знаний студентов - их теоретической готовности к выполнению задания.

При подготовке к зачету в дополнение к изучению конспектов лекций, необходимо пользоваться учебной литературой, рекомендованной настоящей программой. При подготовке к зачету нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала.

Самостоятельная работа студентов (СРС) по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации для обеспечения СРС готовятся преподавателем и выдаются студенту.

Методические материалы и рекомендации для обеспечения самостоятельной работы студентов представлены в методических указаниях для обучающихся по освоению дисциплины.

11. Информационные технологии и программное обеспечение

При выполнении **расчетно-графической работы** студентами предусматривается использование программного обеспечения Microsoft Office (электронные таблицы Microsoft Excel и текстовый редактор Microsoft Word).

А также

1. Виртуальная машина VMware Workstation или VirtualBox.
2. Операционная система Microsoft Windows 10.
3. Среда разработки Microsoft Visual Studio.
4. Среда разработки Qt SDK.
5. Среда веб разработки NetBeans.
6. Среда разработки для языка Prolog SWI Prolog.
7. Среда разработки для языка Haskell Hugs IDE.
8. СУБД MySQL.
9. СУБД SQLite.

12. Описание материально-технической базы необходимой для осуществления образовательного процесса по дисциплине

Лекционные занятия проводятся в обычной аудитории, оснащенной учебной мебелью и доской.

Лабораторные работы по данной дисциплине проводятся в компьютерном классе оборудованным компьютерами с современными лицензионными программно-техническими средствами, с доступом к сети Интернет, столом для конференций.

13. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники и учебные пособия, иная учебная литература, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь.

а) для слабовидящих:

- на экзамене присутствует ассистент, оказывающий студенту необходимую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку);
- задания для выполнения, а также инструкция о порядке проведения зачета/экзамена зачитываются ассистентом;
- письменные задания выполняются на бумаге, надиктовываются ассистенту;
- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;

- студенту для выполнения задания при необходимости предоставляется увеличивающее устройство.

б) для глухих и слабослышащих:

- на экзамене присутствует ассистент, оказывающий студенту необходимую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочесть и оформить задание, в том числе, записывая под диктовку);

- экзамен проводится в письменной форме;

- обеспечивается наличие звукоусиливающей аппаратуры коллективного использования, при необходимости поступающим предоставляется звукоусиливающая аппаратура индивидуального пользования.

- по желанию студента зачет/экзамен может проводиться в письменной форме.

в) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствия верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту.

- по желанию студента экзамен проводится в устной форме

Дополнения и изменения в рабочую программу дисциплины

Внесенные изменения на 20__/20__ учебный год

УТВЕРЖДАЮ

проректор по учебной работе

_____ М.Д. Мукайлов

« ____ » _____ 20 __ г.

В программу дисциплины (модуля) «указывается дисциплина»
по направлению подготовки «указывается направление подготовки (про-
филь)» _____

_____ вносятся следующие изменения:

.....;
.....;
.....;

Программа пересмотрена на заседании кафедры

Протокол № ____ от _____ г.

Заведующий кафедрой

_____/_____/_____
(фамилия, имя, отчество) (ученое звание) (подпись)

Одобрено

Председатель методической комиссии факультета

_____/_____/_____
(фамилия, имя, отчество) (ученое звание) (подпись)

« ____ » _____ 20 __ г.

Лист регистрации изменений в РПД

[illegible]