

год начала подготовки 2018

Документ подписан квалифицированной электронной подписью

Сертификат: 023E519200DAAC0FAC74E9329E4F1A569EE

Владелец: "АНО ВО «РОССИЙСКИЙ НОВЫЙ УНИВЕРСИТЕТ»"; АН

Действител... Елецкий филиал

АНО ВО «Российский новый университет»

**Елецкий филиал Автономной некоммерческой организации высшего образования «Российский новый университет»
(Елецкий филиал АНО ВО «Российский новый университет»)**

кафедра прикладной экономики и сферы обслуживания

Рабочая программа учебной дисциплины (модуля)

Теория алгоритмов

(наименование учебной дисциплины (модуля))

09.03.03 Прикладная информатика

(код и направление подготовки/специальности)

Прикладная информатика в экономике

(код и направление подготовки/специальности, в случаях, если программа разработана для разных направлений подготовки/специальностей)

Рабочая программа учебной дисциплины (модуля) рассмотрена и утверждена на заседании кафедры 12 февраля 2018 г., протокол № 6.

Заведующий кафедрой Прикладной экономики и сферы обслуживания
(название кафедры)

к.п.н., доцент Гнездилова Н.А.

(ученая степень, ученое звание, фамилия и инициалы, подпись заведующего кафедрой)

Елец
2018 год

1. НАИМЕНОВАНИЕ И ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Обеспечение профессионального образования, способствующего социальной, академической мобильности, востребованности на рынке труда, успешной карьере, сотрудничеству.

Формирование у обучающихся систематизированных профессионально значимых знаний по теории алгоритмов и профессиональных умений и навыков, необходимых бакалавру прикладной информатики в экономике.

Изучение учебной дисциплины направлено на получение общих сведений о предмете теория алгоритмов и приобретение теоретических знаний по основным понятиям и методами теории алгоритмов с ориентацией на их использование в практической информатике и вычислительной технике.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП БАКАЛАВРИАТА

Учебная дисциплина Теория алгоритмов относится к вариативной части учебного плана (Б1.В.05).

Содержание учебной дисциплины тесно связано с логикой и содержанием других изучаемых дисциплин:

Учебная дисциплина содержательно и логически связана с другими учебными дисциплинами, изучаемыми студентами:

-предшествует освоению данной дисциплины: Математика, Информационные системы и технологии;

-после изучения данной дисциплины изучается: Математическое и имитационное моделирование, Интеллектуальные информационные системы, Исследование операций и методы оптимизации, Управление информационными системами.

Дисциплина изучается на заочной форме обучения на 3 курсе в 5 семестре.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОП

В результате освоения дисциплины обучающийся должен овладеть следующими компетенциями:

ПК-3. Способность проектировать ИС в соответствии с профилем подготовки по видам обеспечения

Планируемые результаты освоения компетенций

Компетенция	Показатели (планируемые) результаты обучения
<p align="center">ПК-3</p> <p>Способность проектировать ИС в соответствии с профилем подготовки по видам обеспечения.</p>	<p align="center">Владеть:</p> <ul style="list-style-type: none"> - теоретическими и практическими основами проектирования информационных систем В1(ПК-3); - навыками проектирования ИС в соответствии с профилем подготовки по видам обеспечения В2(ПК-3); - навыками осуществления декомпозиции системы на подсистемы и комплексы задач; проектирования компонентов информационного обеспечения, включая, классификаторы, формы и экранные макеты документов, состав и структуру баз данных В3(ПК-3); - навыками использования современных языков и сред программирования, профессионально применяемыми в области проектирования информационных систем, баз данных В4(ПК-3).
	<p align="center">Уметь:</p> <ul style="list-style-type: none"> - проектировать ИС в соответствии с профилем подготовки по видам обеспечения У1(ПК-3); - выполнять выбор средств и методов проектирования отдельных компонент проекта и использовать их при выполнении конкретных работ; использовать международные и отечественные стандарты в области проектирования У2(ПК-3); - адаптировать типовые проектные решения и пакеты прикладных программ, планировать внедрение проекта и осуществлять анализ функционирования и модернизацию систем; разрабатывать планы

	выполнения проектировочных работ У3(ПК-3); - проектировать информационные системы и базы данных с использованием современных CASE-средств, используя функционально-ориентированный и объектно-ориентированный подходы; выбирать и использовать инструментальные средства технологий проектирования У4(ПК-3).
	Знать: - основные принципы проектирования ИС в соответствии с профилем подготовки по видам обеспечения 31(ПК-3); - технологии проектирования ИС, включая технологию типового проектирования, CASE-технологию и технологию быстрого проектирования, и методики обоснования эффективности их применения с учетом социально-культурных технологий 32(ПК-3); - методы и средства структурного и объектно-ориентированного проектирования; методы и средства проектирования БД 33(ПК-3); - методы организации данных, модели предметной области, методы описания процессов в ЭИС 34(ПК-3).

4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ) В ЗАЧЕТНЫХ ЕДИНИЦАХ С УКАЗАНИЕМ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ, ВЫДЕЛЕННЫХ НА КОНТАКТНУЮ РАБОТУ ОБУЧАЮЩИХСЯ С ПРЕПОДАВАТЕЛЕМ (ПО ВИДАМ УЧЕБНЫХ ЗАНЯТИЙ) И НА САМОСТОЯТЕЛЬНУЮ РАБОТУ ОБУЧАЮЩИХСЯ

Дисциплина предполагает изучение 6 тем. Общая трудоемкость дисциплины составляет 4 зачетные единицы (144 часа).

Общий объем учебной дисциплины

№	Форма обучения	Семестр	Общая трудоемкость		В том числе контактная работа с преподавателем					СР	Контроль	
			В з.е.	В часах	Всего	Лекции	Сем	КоР	Конс			Экзамен
1	Заочная	1 сессия, 3 курс	1	36	4	4					32	
		2 сессия, 3 курс	3	108	8		4	1,6	2	0,4	93,4	6,6
	Итого		4	144	12	4	4	1,6	2	0,4	125,4	6,6

Распределение учебного времени по темам и видам учебных занятий заочная форма

№	Наименование разделов, тем учебных занятий	Всего часов	Контактная работа с преподавателем						СР	Контроль	Формируемые результаты обучения
			Всего	Л	Сем	КоР	Конс	Экз			
1	2	3	4	5	6	7	8	9	10	11	12
1.	Интуитивное представление об алгоритмах. Неформальное понятие алгоритма.	9	1	1					8		В1(ПК-3) У1(ПК-3) 31(ПК-3) В2(ПК-3) У2(ПК-3) 32(ПК-3)

2.	Вычислимые функции, разрешимые и перечислимые множества.	9	1	1					8		B1(ПК-3) У1(ПК-3) 31(ПК-3) B2(ПК-3) У2(ПК-3) 32(ПК-3)
3.	Определение машины Тьюринга, Поста. Применение машины Тьюринга к словам. Конструирование машин Тьюринга.	9	1	1					8		B2(ПК-3) B4(ПК-3) У4(ПК-3) У2(ПК-3) 32(ПК-3) 34(ПК-3)
4.	Вычислимые по Тьюрингу функции. Основная гипотеза теории алгоритмов. Машины Тьюринга и современные ЭВМ.	9	1	1					8		B3(ПК-3) У3(ПК-3) 34(ПК-3) B4(ПК-3) У4(ПК-3) 32(ПК-3)
	Итого	36	4	4					32		
5.	Тьюрингов подход к понятию «алгоритм». Алгоритмически разрешимые и неразрешимые проблемы.	11	1		1				10		B1(ПК-3) У1(ПК-3) 31(ПК-3) B2(ПК-3) У2(ПК-3) 32(ПК-3)
6.	Ассоциативные исчисления. Нормальные алгоритмы Маркова. Эквивалентность различных теорий алгоритмов.	11	1		1				10		B1(ПК-3) У1(ПК-3) 31(ПК-3) B2(ПК-3) У2(ПК-3) 32(ПК-3)
7.	Формальная теория вычислимости (частично рекурсивные функции, регистровые машины, машины Тьюринга). Тезис Чёрча. Рекурсивные функции. Тезис Черча.	11	1		1				10		B2(ПК-3) B4(ПК-3) У4(ПК-3) У2(ПК-3) 32(ПК-3) 34(ПК-3)
8.	Неразрешимые алгоритмические проблемы.	11	1		1				10		B3(ПК-3) У3(ПК-3) 34(ПК-3) B4(ПК-3) У4(ПК-3)

											32(ПК-3)
9.	Эффективные операции над вычислимым и функциями.	10							10		B1(ПК-3) B3(ПК-3) У1(ПК-3) У3(ПК-3) 31(ПК-3) 33(ПК-3)
10.	Теорема о неподвижной точке. Общее понятие исчисления. Грамматики.	10							10		B1(ПК-3) У1(ПК-3) 31(ПК-3) B2(ПК-3) B4(ПК-3) У4(ПК-3) У2(ПК-3) 32(ПК-3) 34(ПК-3)
11.	Языки. Иерархия языков по Хомскому. Языки и машины.	10							10		B1(ПК-3) У1(ПК-3) 31(ПК-3) B2(ПК-3) У2(ПК-3) 32(ПК-3)
12.	Основные меры сложности вычисления	10							10		B1(ПК-3) У1(ПК-3) 31(ПК-3) B2(ПК-3) У2(ПК-3) 32(ПК-3)
13.	Приложения теории алгоритмов в информатике.	10							10		B2(ПК-3) B4(ПК-3) У4(ПК-3) У2(ПК-3) 32(ПК-3) 34(ПК-3)
14.	Преобразование символьных данных в компьютере.	3,4							3,4		B3(ПК-3) У3(ПК-3) 34(ПК-3) B4(ПК-3) У4(ПК-3) 32(ПК-3)
	Промежуточная аттестации (экзамен)	10,6				1,6	2	0,4		6,6	
	Итого	108	8		4	1,6	2	0,4	93,4	6,6	
	ИТОГО	144	12	4	4	1,6	2	0,4	125,4	6,6	

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ СТРУКТУРИРОВАННОЕ ПО ТЕМАМ

№ п/п	Наименование раздела, темы учебной дисциплины	Содержание раздела, темы
1	2	3
1. ___	Интуитивное представление об алгоритмах. Алгоритмы в математике.	Интуитивное представление об алгоритмах. Неформальное понятие алгоритма. Свойства алгоритмов. Способы записи алгоритмов. Основные алгоритмические структуры.

		<p>Литература: Обязательная: 1-2. Дополнительная: 1-3.</p>
2.	Машины Тьюринга и Поста. Нормальный алгоритм Маркова.	<p>Необходимость уточнения понятия алгоритма. Формальное описание алгоритма. Определение машины Тьюринга. Конструирование машин Тьюринга. Машина Поста. Программы для машины Поста. Марковские подстановки. Нормальные алгоритмы Маркова.</p> <p>Литература: Обязательная: 1-2. Дополнительная: 1-3.</p>
3.	Алгоритмически неразрешимые проблемы и вычислимые функции	<p>Десятая проблема Гильберта. Проблема останова. Невозможность создания универсального алгоритма отладки программы. Теорема Чёрча. Определение вычислимой функции.</p> <p>Литература: Обязательная: 1-2. Дополнительная: 1-3.</p>
4.	Рекурсивные функции	<p>Происхождение рекурсивных функций. Операторы суперпозиции, примитивной рекурсии и минимизации. Примитивно рекурсивные функции. Частично рекурсивные и общерекурсивные функции. Тезис Чёрча.</p> <p>Литература: Обязательная: 1-2. Дополнительная: 1-3.</p>
5.	Понятие сложности алгоритма	<p>Проблема выбора алгоритма. Определение вычислительного процесса. Временная сложность алгоритма. Теоретическая сложность алгоритма. Эффективность алгоритма.</p> <p>Литература: Обязательная: 1-2. Дополнительная: 1-3.</p>
6.	Анализ алгоритмов поиска и сортировки	<p>Последовательный поиск в неупорядоченном массиве. Алгоритм бинарного поиска в упорядоченном массиве. Определение сортировки. «Пузырьковый» метод сортировки. Сортировка выбором. Сортировка вставками. Сортировка слиянием.</p> <p>Литература: Обязательная: 1-2. Дополнительная: 1-3.</p>

Планы практических занятий

Тема 1. Представление об алгоритмах. Алгоритмы в математике.

1. Интуитивное представление об алгоритмах.

Интуитивное представление об алгоритмах. Неформальное понятие алгоритма.

Свойства алгоритмов.

2. Алгоритмы в математике.

Способы записи алгоритмов. Основные алгоритмические структуры.

Тема 2. Машины Тьюринга и Поста. Нормальный алгоритм Маркова.

1. Машины Тьюринга и Поста.

Необходимость уточнения понятия алгоритма. Формальное описание алгоритма.

Определение машины Тьюринга. Конструирование машин Тьюринга. Машина Поста. Программы для машины Поста.

2. Нормальный алгоритм Маркова.

Марковские подстановки. Нормальные алгоритмы Маркова.

Тема 3. Алгоритмически неразрешимые проблемы и вычислимые функции.

Десятая проблема Гильберта. Проблема останова. Невозможность создания универсального алгоритма отладки программы. Теорема Чёрча. Определение вычислимой

функции.

Тема 4. Рекурсивные функции.

Происхождение рекурсивных функций. Операторы суперпозиции, примитивной рекурсии и минимизации. Примитивно рекурсивные функции. Частично рекурсивные и общерекурсивные функции. Тезис Чёрча.

Тема 5. Понятие сложности алгоритма.

Проблема выбора алгоритма. Определение вычислительного процесса. Временная сложность алгоритма. Теоретическая сложность алгоритма. Эффективность алгоритма.

Тема 6. Анализ алгоритмов поиска и сортировки.

Последовательный поиск в неупорядоченном массиве. Алгоритм бинарного поиска в упорядоченном массиве. Определение сортировки. «Пузырьковый» метод сортировки. Сортировка выбором. Сортировка вставками. Сортировка слиянием.

6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Контроль самостоятельной работы студента осуществляется в форме:

изучения:

- первоисточников,
- дат и событий,
- терминологии.

ответов:

- на вопросы для самопроверки,

подготовки:

- сообщений,
- рефератов,
- презентаций.

решений:

- заданий,
- тестов.

6.1. Задания для приобретения, закрепления и углубления знаний.

6.1.1 Основные категории учебной дисциплины для самостоятельного изучения:

Алгоритм (процедура) – решение задач в виде точных последовательно выполняемых предписаний.

Это интуитивное определение сопровождается описанием интуитивных свойств (признаков) алгоритмов: эффективность, определенность, конечность.

Эффективность – возможность исполнения предписаний за конечное время.

Определенность – возможность точного математического определения или формального описания содержания команд и последовательности их применения в этой процедуре.

Конечность – выполнение алгоритма при конкретных исходных данных за конечное число шагов.

В формальных описаниях алгоритм конструктивно связывают с понятием машины, предназначенной для автоматизированных преобразований символьной информации.

Общая Теория алгоритмов занимается проблемой эффективной вычислимости. моделями алгоритмических преобразований символьной информации являются:

- конечные автоматы;
- машина Тьюринга;
- машина Поста;
- ассоциативное исчисление или нормальные алгоритмы Маркова;
- рекурсивные функции.

Верификация (от лат. verus – «истинный» и facere – «делать») – проверка, способ

подтверждения каких-либо теоретических положений, алгоритмов, программ и процедур путем их сопоставления с опытными (эталонными или эмпирическими) данными, алгоритмами и программами.

Тестирование применяется для определения соответствия предмета испытания заданным спецификациям.

Алфавит языка обозначается как конечное множество символов.

Например: $\Sigma = \{a, b, c, d\}$, $\Sigma = \{0, 1\}$.

Символ и цепочка символов образуют слово – $a, b, 0, abcd, 0111000$. Пустое слово (ϵ) не содержит символов. Множество слов $S = \{a, ab, aaa, bc\}$ в алфавите Σ называют языком $L(\Sigma)$.

Символы алфавита могут соединяться конкатенацией (сцепление, соединение) в цепочки символов-слов, которые соединяются в новые слова.

Конкатенация двух слов $x|y$ обозначает, что к слову x справа приписано слово y или $x|y = xy$, причем $xy \neq yx$.

Объединение $(S1 \cup S2)$ или $(S1 + S2)$ множеств $S1 = \{a, aa, ba\}$, $S2 = \{e, bb, ab\}$, $S1 \cup S2 = \{a, aa, ba, e, bb, ab\}$.

Для операции объединения выполняются следующие законы:

Коммутативность объединения $S1 \cup S2 = S2 \cup S1$.

Идемпотентность объединения $S \cup S = S$.

Ассоциативность объединения $S1 \cup (S2 \cup S3) = (S1 \cup S2) \cup S3$.

Дистрибутивность конкатенации (умножения) и объединения

$S1(S2 \cup S3) = S1S2 \cup S1S3$.

Итерация множества $\{S\}^*$ состоит из пустого слова и всех слов вида $S^0 = \epsilon$, $S^1 = S$, $S^2 = SS$, $S^3 = SSS$.

Формулы, содержащие эти операции с множествами слов, называют регулярными выражениями.

Ассоциативность итерации $S1^* (S2^* S3) = (S1^* S2)^* S3$.

Дистрибутивность объединения с итерацией $S1^* (S2 \cup S3) = S1^* S2 \cup S1^* S3$.

Языки, определяемые регулярными выражениями, называются регулярными языками, а множество слов – регулярными множествами.

Алгоритм распознавания предложений регулярного языка называют конечным автоматом (КА).

Определение. Конечный автомат определяется символами $M = (Q, \Sigma, \delta, q_0, F)$, где

$Q = \{q_0, q_1, \dots, q_n\}$ – конечное множество состояний;

$\Sigma = \{a, b, c, \dots\}$ – входной алфавит (конечное множество);

$\delta: Q \times \Sigma \rightarrow \{P_j\}$ – функция переходов, P_j – подмножество Q .

Конечный автомат называется недетерминированным (НДКА), если P_j содержит более одного состояния.

КА называется детерминированным (ДКА), если P_j содержит не более одного состояния.

КА полностью определен, если P_j в детерминированном автомате не пусто. Если есть пустые элементы множества P_j , то автомат частично определен.

Конфигурация КА $k = (q, \omega)$, где q – текущее состояние КА, ω – неп прочитанная цепочка символов слова на ленте, включая символ под читающей головкой.

$k = (q, \omega)$ – текущая конфигурация;

$k_0 = (q_0, \omega_0)$ – начальная конфигурация;

$k_f = (q, \epsilon)$, где $q \in F$, – заключительная конфигурация и (ϵ) – символ, обозначающий конец строки.

Шаг алгоритма – переход из одной конфигурации КА в другую $K_i \rightarrow K_j$ или $(q_i, \omega_i) \rightarrow (q_j, \omega_j)$.

Исполнение алгоритма это последовательность шагов, в которых изменяется конфигурация КА: $(p, 01001) \rightarrow (q, 1001) \rightarrow (p, 001) \rightarrow (q, 01) \rightarrow (r, 1) \rightarrow (r, \epsilon)$;

(р, 01001)- начальная конфигурация;

(r, e) – конечная конфигурация.

Лемма. Число состояний в ДКА не превышает $2n-1$, где n – число состояний в НДКА.

Лемма о накачке (pumping): Пусть L – регулярный язык. Тогда существует константа n , для которой каждое слово языка L длиной $m \geq n$ можно разделить на три слова xuz так, что длина $xu \leq \leq \leq n$ и для любого $k \geq 0$ слово xuk^2z принадлежит L .

Событиями являются предикаты:

- идентификации и выбора состояний $A=(is A)$;

- идентификации входов $a=(is a)$;

- выбора выходов $\alpha=(is \alpha)$.

Универсальный КА, применяемый для решения любой алгоритмически разрешимой задачи, в теории алгоритмов и вычислений называется машиной Тьюринга.

Основная гипотеза теории алгоритмов: Машина Тьюринга решает любую алгоритмически разрешимую задачу.

Вопрос алгоритмической разрешимости (существования алгоритма решения задачи) сводится к доказательству существования машины T , решающей задачу за конечное число шагов. Если число шагов бесконечно, то задача трудно разрешимая или алгоритмически неразрешимая проблема.

Тезис Тьюринга (неформальный): Каждую вычислимую (программой какого угодно языка программирования) функцию типа $\Sigma^* \rightarrow \Sigma^*$ можно вычислить на подходящей машине Тьюринга.

Конструктивно машина Поста близка к машине Тьюринга. Принципиально она отличается двоичным алфавитом входных и выходных данных (в машине Тьюринга алфавит не определен, а выбирается).

В машине Тьюринга следует сконструировать управляющий автомат, а в машине Поста – программу решения задачи.

Ассоциативное исчисление – совокупность слов в алфавите и схема алгоритма – конечная система подстановок.

Алгоритм в ассоциативном исчислении – формальное решение задачи распознавания эквивалентности слов $A \sim Q$ с использованием подстановок.

Схему применения подстановок, выполненную в виде читающего и записывающего автомата, которая формирует символическую строку команд, называют компилятором.

Рекурсивные (возвратные) функции позволяют определить значение от неизвестного к известному (от сложного к простому).

Функция вычислима, если существует алгоритм – эффективная последовательная процедура вычисления от простого к сложному.

Выделяется базис элементарных функций, интуитивно вычисляемых, и средства-операторы получения из них более сложных функций.

Классы сложности – множества вычислительных задач, примерно одинаковых по сложности вычисления. Более узко, классы сложности — это множества предикатов (функций, получающих на вход слово и возвращающих ответ 0 или 1), использующих для вычисления примерно одинаковые количества ресурсов.

Ряд алгоритмов имеют порядок, включающий $\log_2 n$, и называются логарифмическими (logarithmic).

Алгоритмы, имеющие порядок $O(n^2)$, являются квадратичными (quadratic).

Алгоритм показывает кубическое (cubic) время, если его порядок равен $O(n^3)$, и такие алгоритмы очень медленные.

Алгоритм Флойда – Уоршелла (динамический алгоритм для нахождения кратчайших расстояний между всеми вершинами взвешенного ориентированного графа).

Алгоритм со сложностью $O(2^n)$ имеет экспоненциальную сложность (exponential complexity). Такие алгоритмы выполняются настолько медленно, что они используются только при малых значениях n .

Алгоритмы со сложностью $O(n!)$ – факториальные алгоритмы, в основном, используются в комбинаторике для определения числа сочетаний, перестановок.

В основах теории алгоритмов рассматриваем задачи, для которых возможен

переход от содержательной (вербальной) формулировки к алгоритмическому решению.

Алгоритм представлен в формальной записи – конечными автоматами, синтаксическим разбором, рекурсивными функциями, которые очевидно преобразуются в программы вычислительных машин.

Скомпилированные по этим моделям программы ВМ не требуют верификации и, по существу, построение этих моделей является доказательством существования алгоритма.

Модели алгоритмов используются в алгоритмических языках, и возможность их реализации является признаком универсальности языка.

В практическом программировании чаще эти модели не применяются, используется процедурное, объектное и функциональное расширения, которые уже относятся к инженерии программирования.

6.2. Задания для повторения и углубления приобретаемых знаний.

Задание 6.2.1. 31(ПК-3) *Представление об алгоритмах. Алгоритмы в математике.*

1. Дайте определение интуитивному представлению об алгоритмах.
2. Поясните неформальное понятие алгоритма.
3. Перечислите свойства алгоритмов.
4. Расскажите про способы записи алгоритмов.
5. Поясните основные алгоритмические структуры.

Задание 6.2.2 32(ПК-3) *Машины Тьюринга и Поста. Нормальный алгоритм Маркова.*

1. Обоснуйте необходимость уточнения понятия алгоритма.
2. Объясните формальное описание алгоритма.
3. Дайте определение машины Тьюринга.
4. Поясните конструирование машин Тьюринга.
5. Что такое Машина Поста?
6. Расскажите о программах для машины Поста.
7. Что такое Марковские подстановки?
8. Определите нормальные алгоритмы Маркова.

Задание 6.2.3 33 (ПК-3) *Алгоритмически неразрешимые проблемы и вычислимые функции.*

1. Поясните десятую проблему Гильберта.
2. Что такое Проблема останова?
3. Поясните невозможность создания универсального алгоритма отладки программы.
4. Сформулируйте Теорему Чёрча.
5. Дайте определение вычислимой функции.

Задание 6.2.4 34(ПК-3) *Рекурсивные функции.*

1. Поясните происхождение рекурсивных функций.
2. Определите операторы суперпозиции, примитивной рекурсии и минимизации.
3. Определите примитивно рекурсивные функции.
4. Определите частично рекурсивные и общерекурсивные функции.
5. Поясните Тезис Чёрча.

Задание 6.2.5 31(ПК-3) *Понятие сложности алгоритма.*

1. Поясните проблему выбора алгоритма.
2. Дайте определение вычислительного процесса.

Задание 6.2.6 32(ПК-3) *Понятие сложности алгоритма.*

1. Что такое Временная сложность алгоритма?
2. Что такое Теоретическая сложность алгоритма?
3. Поясните понятие Эффективность алгоритма.

Задание 6.2.7 33(ПК-3) *Анализ алгоритмов поиска.*

1. Поясните последовательный поиск в неупорядоченном массиве.
2. Объясните алгоритм бинарного поиска в упорядоченном массиве.

Задание 6.2.8 34(ПК-3) *Анализ алгоритмов сортировки.*

1. Дайте определение сортировки.
2. Определите «Пузырьковый» метод сортировки.
3. Поясните сортировку выбором.
4. Поясните сортировку вставками.
5. Поясните сортировку слиянием.

6.3. Задания, направленные на формирование профессиональных умений:

Задание 6.3.1. У1(ПК-3)

Подготовьте реферат на тему «Алгоритмы в математике».

Задание 6.3.2. У2(ПК-3)

Составьте презентацию «Машины Тьюринга и Поста».

Задание 6.3.3. У3 (ПК-3)

Подготовьте эссе на тему «Алгоритмически неразрешимые проблемы и вычислимые функции».

Задание 6.3.4. У4(ПК-3)

Подготовьте реферат на тему «Рекурсивные функции».

Задание 6.3.5. У1(ПК-3)

Составьте презентацию «Проблемы выбора алгоритма».

Задание 6.3.6. У2(ПК-3)

Подготовьте реферат на тему «Эффективность алгоритма».

Задание 6.3.7. У3(ПК-3)

Составьте презентацию «Анализ алгоритмов поиска».

Задание 6.3.8. У4(ПК-3)

Подготовьте эссе на тему «Анализ алгоритмов сортировки».

6.4. Задания, направленные на формирование профессиональных навыков, владений

Задание 6.4.1. В1(ПК-3)

Пусть семейство состоит из языков С и PASCAL, языки имеют полное описание их синтаксиса и операционной семантики. Языки обладают компиляторами в ASSEMBLER, поэтому для программирования б) $b/\alpha, R a/\alpha, R a/\beta, R a/\beta, R b/\gamma, R q_0 q_1 q_2$ е е е $b/\alpha a/\alpha a/\beta a/\beta b/\gamma q_0 q_1 q_2$ а) соответствующего частного случая Тезиса Тьюринга требуется построить компилятор, преобразующий ассемблерный код в программу для машины Тьюринга. Последнее является весьма трудоемкой, но реалистичной задачей

по программированию. Для полного математического доказательства потребуется еще верифицировать программу на языке.

Задание 6.4.2. В2(ПК-3)

1. Составить программу для машины Тьюринга «сложение» $q_1 \rightarrow q_0$, $q_1: \dots 0001^n 01^m 000 \dots$, $q_2: \dots 0001^{n+m} 0000 \dots$.

2. Составить программу для машины Поста «сложение» $q_1 \rightarrow q_0$, $q_1: \dots 0001^n 01^m 000 \dots$, $q_2: \dots 0001^{n+m} 0000 \dots$.

Задание 6.4.3. В3(ПК-3)

1. Составить нормальный алгоритм Маркова, описывающий сложение двух натуральных чисел (представленных наборами единиц, разделенных символом «+»). Алфавит: $\{+, 1\}$.

Задание 6.4.4 В4(ПК-3)

1. Напишите программу для машины Поста, которая стирает крайние метки в слове произвольной длины. Начальное расположение каретки – над пустой ячейкой слева от слова. Синтаксис: Л – каретка влево; П – каретка вправо; V – метка; X – стирание метки; ? x, y – если ячейка пустая, то переход на строку x, иначе на строку y; ! – конец работы.
2. Напишите программу для машины Тьюринга, которая прибавляет 10 к произвольному десятичному числу. Начальное положение каретки – над крайним правым символом числа. Алфавит $A = \{a0, 0, 1, 2 \dots 9\}$. $T = \{Л, П, Н\}$ – множество действий каретки (влево, вправо, остаться на месте).

Задание 6.4.5 В1(ПК-3)

3. Напишите программу для машины Поста, которая добавляет 2 метки справа к слову произвольной длины. Начальное расположение каретки – над крайней левой меткой слова. Синтаксис: Л – каретка влево; П – каретка вправо; V – метка; X – стирание метки; ? x, y – если ячейка пустая, то переход на строку x, иначе на строку y; ! – конец работы.
4. Напишите программу для машины Тьюринга, которая вычитает 1 из произвольного положительного десятичного числа. Начальное положение каретки – над крайним правым символом числа. Алфавит $A = \{a0, 0, 1, 2 \dots 9\}$. $T = \{Л, П, Н\}$ – множество действий каретки (влево, вправо, остаться на месте).

Задание 6.4.6 В2(ПК-3)

5. Напишите программу для машины Поста, которая стирает 2 метки справа у слова произвольной длины. Начальное расположение каретки – над крайней левой меткой слова. Синтаксис: Л – каретка влево; П – каретка вправо; V – метка; X – стирание метки; ? x, y – если ячейка пустая, то переход на строку x, иначе на строку y; ! – конец работы.
6. Напишите программу для машины Тьюринга, которая заменяет символы 0 на 1 и 1 на 0 в произвольном слове. Начальное положение каретки – слева от слова. Алфавит $A = \{a0, 0, 1\}$. $T = \{Л, П, Н\}$ – множество действий каретки (влево, вправо, остаться на месте). Пример слова: 001101 (Конечный результат – 110010).

Задание 6.4.7 В3(ПК-3)

7. Напишите программу для машины Поста, которая добавляет по одной метке справа и слева к слову произвольной длины. Начальное расположение каретки – над крайней правой меткой слова. Синтаксис: Л – каретка влево; П – каретка вправо; V – метка; X – стирание метки; ? x, y – если ячейка пустая, то переход на строку x, иначе на строку y; ! – конец работы.
8. Напишите программу для машины Тьюринга, которая заменяет в произвольном слове все единицы звездочками. Начальное положение каретки – слева от слова. Алфавит $A = \{a0, 1, 2, 3, *\}$. $T = \{Л, П, Н\}$ – множество действий каретки (влево, вправо, остаться на месте).

Задание 6.4.8 В4(ПК-3)

Построить граф автомата, регулярное выражение, систему уравнений, интерпретирующих КА, заданный таблицей переходов и восстановить исходное регулярное выражение, выполнить преобразование в блок-схему и систему уравнений перехода.

Соотношение заданий с формируемыми показателями обучения

Формируемая компетенция	Показатели сформированности компетенции	Задания, направленные на: - приобретение новых знаний, углубления и закрепления ранее приобретенных знаний; - формирование профессиональных умений и навыков
	Владеть: - теоретическими и практическими основами проектирования информационных систем В1(ПК-3); - навыками проектирования ИС в соответствии с	Задание 6.4.1. В1(ПК-3) Задание 6.4.2. В2(ПК-3) Задание 6.4.3. В3(ПК-3) Задание 6.4.4 В4(ПК-3)

<p>ПК-3 Способность проектировать ИС в соответствии с профилем подготовки по видам обеспечения.</p>	<p>профилем подготовки по видам обеспечения В2(ПК-3);</p> <ul style="list-style-type: none"> - навыками осуществления декомпозиции системы на подсистемы и комплексы задач; проектирования компонентов информационного обеспечения, включая, классификаторы, формы и экранные макеты документов, состав и структуру баз данных В3(ПК-3); - навыками использования современных языков и сред программирования, профессионально применяемыми в области проектирования информационных систем, баз данных В4(ПК-3). 	<p>Задание 6.4.5 В1(ПК-3) Задание 6.4.6 В2(ПК-3) Задание 6.4.7 В3(ПК-3) Задание 6.4.8 В4(ПК-3)</p>
	<p>Уметь:</p> <ul style="list-style-type: none"> - проектировать ИС в соответствии с профилем подготовки по видам обеспечения У1(ПК-3); - выполнять выбор средств и методов проектирования отдельных компонент проекта и использовать их при выполнении конкретных работ; использовать международные и отечественные стандарты в области проектирования У2(ПК-3); - адаптировать типовые проектные решения и пакеты прикладных программ, планировать внедрение проекта и осуществлять анализ функционирования и модернизацию систем; разрабатывать планы выполнения проектировочных работ У3(ПК-3); - проектировать информационные системы и базы данных с использованием современных CASE-средств, используя функционально-ориентированный и объектно-ориентированный подходы; выбирать и использовать инструментальные средства технологий проектирования У4(ПК-3). 	<p>Задание 6.3.1. У1(ПК-3) Задание 6.3.2. У2(ПК-3) Задание 6.3.3. У3 (ПК-3) Задание 6.3.4. У4(ПК-3) Задание 6.3.5. У1(ПК-3) Задание 6.3.6. У2(ПК-3) Задание 6.3.7. У3(ПК-3) Задание 6.3.8. У4(ПК-3)</p>
	<p>Знать:</p> <ul style="list-style-type: none"> - основные принципы проектирования ИС в соответствии с профилем подготовки по видам обеспечения З1(ПК-3); - технологии проектирования ИС, включая технологию типового проектирования, CASE-технологию и технологию быстрого проектирования, и методики обоснования эффективности их применения с учетом социально-культурных технологий З2(ПК-3); - методы и средства структурного и объектно-ориентированного проектирования; методы и средства проектирования БД З3(ПК-3); - методы организации данных, модели предметной области, методы описания процессов в ЭИС З4(ПК-3). 	<p>Задание 6.2.1. З1(ПК-3) Задание 6.2.2 З2(ПК-3) Задание 6.2.3 З3(ПК-3) Задание 6.2.4 З4(ПК-3) Задание 6.2.5 З1(ПК-3) Задание 6.2.6 З2(ПК-3) Задание 6.2.7 З3(ПК-3) Задание 6.2.8 З4(ПК-3)</p>

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Средства оценивания в ходе текущего контроля:

7.1.1 Задания для оценки знаний

7.1.1.1 Тестовые задания (ПК-3)

Вариант I.

№	Вопрос	Вар-ты ответа	Ответ
---	--------	---------------	-------

1. Вычислительный процесс — это
 1. последовательность действий, выполняемых исполнителем
 2. порядок выполнения алгоритма в применении к исходным данным
 3. последовательность шагов алгоритма
 4. совокупность промежуточных значений переменных1

2. Исходные данные — это
 1. точно определенное множество значений, с которых начинается выполнение алгоритма
 2. переменные и константы, которые используются в алгоритме
 3. множество возможных значений переменных
 4. набор всех переменных алгоритма и их значений1

3. Состоянием вычислительного процесса, порожденного алгоритмом А называют
 1. состояние на множестве переменных (набор всех переменных, используемых в алгоритме А и их значение всех переменных в данный момент времени)
 2. точно определенное множество значений, с которых начинается выполнение алгоритма
 3. множество возможных значений переменных
 4. совокупность значений переменных из терминального состояния вычислительного процесса алгоритма.1

4. Терминальным состоянием вычислительного процесса является
 1. состояние, на множестве значений которого выполняется определенное условие — правило Окончания алгоритма.
 2. состояние на множестве переменных (набор всех переменных, используемых в алгоритме А и их значение всех переменных в данный момент времени)
 3. множество возможных значений переменных
 4. переход из одного состояния в другое1.

5. Результат — это
 1. определенная совокупность значений из терминального состояния вычислительного процесса алгоритма
 2. состояние, на множестве значений которого выполняется определенное условие — правило окончания алгоритма.
 3. значения выходных переменных в данный момент времени
 4. множество возможных значений результирующих переменных

7. Какие из перечисленных свойств алгоритма являются необходимыми
 1. дискретность
 2. результативность
 3. массовость
 4. детерминированность
 5. рекурсивность
 6. формальность и простота
 7. эффективность1
2
4

8. Какие из перечисленных свойств алгоритма являются сравнительными
 1. дискретность
 2. результативность
 3. массовость
 4. детерминированность
 5. рекурсивность
 6. формальность и простота
 7. эффективность3

- | | | |
|----|---|---|
| | | 5 |
| | | 6 |
| | | 7 |
| 9. | Какие из функций являются базовыми рекурсивными | |
| | 1. функции любого числа независимых переменных, тождественно равные нулю. | |
| | 2. функции любого числа независимых переменных, тождественно равные одному из аргументов. | |
| | 3. функции получения последователя одного независимого переменного. | |
| | 4. функция, полученная с помощью оператора суперпозиции | |
| | 5. функция, полученная с помощью оператора минимизации | |
| | 6. любые всюду вычислимые функции | 1 |
| | | 2 |
| | | 3 |
| 10 | Какие функции не являются общерекурсивными | |
| | 1. функции любого числа независимых переменных, тождественно равные нулю. | |
| | 2. функции любого числа независимых переменных, тождественно равные одному из аргументов. | |
| | 3. функции получения последователя одного независимого переменного. | |
| | 4. функция, полученная с помощью оператора суперпозиции из рекурсивных функций | |
| | 5. функция, полученная с помощью оператора минимизации из рекурсивных функций | |
| | 6. функции, полученные с помощью оператора примитивной рекурсии из рекурсивных функций | 5 |
| 11 | Алгоритм построения суперпозиции двух функций | |
| | 1. Выбираются две функции f, g . | |
| | 2. Определяется аргумент x_K первой функции f , для которой будем осуществлять подстановку. | |
| | 3. Подставляем значение аргумента в g и вычисляем её значение g_0 . | |
| | 4. $x_K = G_0$. | |
| | 5. $f_0 = F(x_1, \dots, x_K, \dots, x_M)$ | |
| 13 | Алгоритм оператора построения по первому нулю | |
| | 1) фиксируем значение переменных x_1, x_2, \dots, x_N ; | |
| | 2) строим некую функцию f , к которой будет добавлена переменная x_K ; | |
| | 3) определяем, имеет ли относительно этой переменной функция f натуральный корень. Если корней нет, то при данных значениях функция g не определена. Если корни есть, то находим минимальный корень u . Этот корень и есть искомое значение функции g , $g = u$. | |
| 14 | В блок-схеме описана команда | |
| | 1. цикла с предусловием | |
| | 2. цикла с параметром | |
| | 3. полного ветвления | |
| | 4. неполного ветвления | 3 |
| 15 | В блок-схеме описана команда | |
| | 1. цикла с предусловием | |
| | 2. цикла с параметром | |
| | 3. полного ветвления | |
| | 4. неполного ветвления | 1 |
| 16 | В блок-схеме описана команда | |
| | 1. цикла с предусловием | |
| | 2. цикла с параметром | |
| | 3. полного ветвления | |

4. неполного ветвления 4
- 17 В блок схеме описана команда
 1. цикла с предусловием
 2. цикла с параметром
 3. полного ветвления
 4. неполного ветвления 2
- 18 В блок схеме описана команда
 1. цикла с предусловием
 2. цикла с параметром
 3. цикла с постусловием
 4. неполного ветвления 3
- 19 Алгоритм – это
 1. понятное и точное описание конечной последовательности команд, приводящей от исходных данных к искомому результату
 2. пошаговое описание процесса решения какой-либо задачи
 3. последовательность действий, применяемая к некоторым исходным данным 1
- 20 Свойство, означающее, что процесс решения задачи, определяемый алгоритмом, расчленен на отдельные элементарные шаги, соответствует
 1. дискретности
 2. детерминированности
 3. результативности
 4. массовости 1
- 21 Существование на каждом шаге алгоритма однозначного выбора и отсутствие неоднозначных конструкций соответствует свойству
 1. дискретности
 2. детерминированности
 3. результативности
 4. массовости 2
- 22 Свойство, означающее, что алгоритм всегда приводит к результату через конечное число шагов, соответствует
 1. результативности
 2. дискретности
 3. определенности
 4. массовости 1
- 23 $?3,4(x, y, z)=$
 1. x
 2. y
 3. z
 4. значение не определено 4
- 24 $? (? (?(x)))=$
 1. $x+3$
 2. $x+x+x$
 3. $x+1$ 1
- 25 Класс данных, к которым применим данный алгоритм, определяет
 1. массовость
 2. результативность
 3. эффективность
 4. рекурсивность

№	Показатели сформированности компетенции	ФОС текущего контроля (тестовые задания)
1.	31(ПК-3)	1-25

2.	32(ПК-3).	1-25
3.	33(ПК-3).	1-25
4.	34(ПК-3).	1-25
5.	31(ПК-3).	1-25
6.	32(ПК-3).	1-25
7.	33(ПК-3).	1-25
8.	34(ПК-3).	1-25

7.1.2 Задания для оценки умений
7.1.2.1 Примерные темы сообщений (ПК-3)

Сообщения (устная форма) позволяет глубже ознакомиться с отдельными, наиболее важными и интересными процессами, осмыслить, увидеть их сложность и особенности.

1. Машина Тьюринга.
2. Машина Поста.
3. Тезис Поста.
4. Нормальные алгоритмы Маркова.
5. Алгоритмически неразрешимые задачи.
6. Вычислимые функции.
7. Рекурсивные функции.
8. Операторы суперпозиции, примитивной рекурсии и минимизации.
9. Операторы суперпозиции, примитивной рекурсии и минимизации.
10. Примитивно рекурсивные функции.
11. Частично рекурсивные и общерекурсивные функции.
12. Понятие сложности алгоритма
13. Временная сложность алгоритма.
14. Теоретическая сложность алгоритма.
15. Эквивалентность различных теорий алгоритмов.
16. Машины Тьюринга и современные электронно-вычислительные машины.

№	Показатели сформированности компетенции	ФОС текущего контроля (тематика сообщений)
1.	У1(ПК-3)	1-16
2.	У2(ПК-3)	1-16
3.	У3(ПК-3)	11, 13, 15, 16
4.	У4(ПК-3)	1-16
5.	У1(ПК-3)	1-16
6.	У2(ПК-3)	9-16
7.	У3(ПК-3)	1-16
8.	У4(ПК-3)	1-16

7.1.2.2 Темы рефератов (ПК-3)

1. Машина Тьюринга.
2. Тезис Тьюринга.
3. Машина Поста.
4. Тезис Поста.
5. Нормальные алгоритмы Маркова.
6. Алгоритмически неразрешимые задачи.
7. Теорема Чёрча.
8. Вычислимые функции.
9. Происхождение рекурсивных функций.
10. Операторы суперпозиции, примитивной рекурсии и минимизации.

11. Операторы суперпозиции, примитивной рекурсии и минимизации.
12. Примитивно рекурсивные функции.
13. Частично рекурсивные и общерекурсивные функции.
14. Понятие сложности алгоритма
15. Временная сложность алгоритма.
16. Теоретическая сложность алгоритма.
17. Эквивалентность различных теорий алгоритмов.
18. Машины Тьюринга и современные электронно-вычислительные машины.

№	Показатели сформированности компетенции	ФОС текущего контроля (тематика рефератов)
1.	У1(ПК-3)	1-18
2.	У2(ПК-3)	1-18
3.	У3(ПК-3)	1-18
4.	У4(ПК-3)	1-18
5.	У1(ПК-3)	1-18
6.	У2(ПК-3)	1-18
7.	У3(ПК-3)	1-18
8.	У4(ПК-3)	1-18

7.1.2.3. Примерная тематика презентаций (ПК-3)

Презентация – набор слайдов в Power Point. Выступление по презентации не требуется и оценивается дополнительно.

Преподаватель каждый раз выбирает самостоятельно количество слайдов (в зависимости от количества учебных часов по дисциплине) от 10 слайдов и до 30 по одной проблематике.

Название документа – ФИО студента (Иванов И.П.ppt);

Первый слайд – тема презентации, далее – сам материал. План, актуальность темы, введение, заключение и список литературы не являются составной частью презентации и

делаются студентом по собственному желанию.

Презентация в обязательном порядке включает следующие элементы:

- картинки и фото;
- графические элементы;
- классификации;
- таблицы;
- логические цепочки;
- схемы;
- выводы.

Ссылка при цитировании на источник в презентации обязательна. Все данные должны быть сопровождаемы годами.

Презентации на темы:

1. Алгоритмы в математике
2. Машины Тьюринга и Поста.
3. Алгоритмически неразрешимые проблемы и вычислимые функции.
4. Рекурсивные функции.
5. Проблемы выбора алгоритма
6. Эффективность алгоритма.
7. Анализ алгоритмов поиска
8. Анализ алгоритмов сортировки.

№	Показатели сформированности компетенции	ФОС итогового контроля (тематика презентаций)
1.	У1(ПК-3)	1-8
2.	У2(ПК-3)	2, 3, 4, 6
3.	У3(ПК-3)	1-8
4.	У4(ПК-3)	1-8
5.	У1(ПК-3)	1-8
6.	У2(ПК-3)	2-8
7.	У3(ПК-3)	1-8
8.	У4(ПК-3)	1-8

7.1.3 Задания для оценки навыков, владений, опыта деятельности

7.2.3.1 Задачи по дисциплине (ПК-3)

1. Составить программу для машины Тьюринга «сложение» $q_1 \rightarrow q_0$, $q_1: \dots 0001^n 01^m 000 \dots$, $q_2: \dots 0001^{n+m} 0000 \dots$.
2. Составить программу для машины Поста «сложение» $q_1 \rightarrow q_0$, $q_1: \dots 0001^n 01^m 000 \dots$, $q_2: \dots 0001^{n+m} 0000 \dots$.
3. Составить программу для машины Тьюринга «удвоение» $q_1 \rightarrow q_0$, $q_1: \dots 0001^n 000 \dots$, $q_2: \dots 0001^{2n} 000 \dots$.
4. Составить программу для машины Поста «вычитание единицы» $q_1 \rightarrow q_0$, $q_1: \dots 0001^n 000 \dots$, $q_2: \dots 0001^{n-1} 000 \dots$.
5. Составить нормальный алгоритм Маркова «сортировка по алфавиту» (алфавит $\{a,b,c\}$).
6. Составить нормальный алгоритм Маркова, который удаляет из строки, состоящей из букв a и b , первый символ.
7. Составить программу для машины Поста «увеличение массива на 2 символа» $q_1 \rightarrow q_0$, $q_1: \dots 0001^n 000 \dots$, $q_2: \dots 0001^{n+2} 000 \dots$.
8. Составить программу для машины Тьюринга «перенос нуля» $q_1 \rightarrow q_0$, $q_1: \dots 0001^n 0000 \dots$, $q_2: \dots 00001^n 000 \dots$.
9. Составить программу для машины Тьюринга «прибавление единицы» для алфавита $\{0,1,2,3,4,5,6,7,8,9\}$.
10. Составить программу для машины Тьюринга «вычитание единицы» для алфавита $\{0,1,2,3,4,5,6,7,8,9\}$.
11. Составить программу для машины Поста, которая удаляет 2 крайних элемента массива.
12. Составить нормальный алгоритм Маркова, описывающий сложение двух натуральных чисел (представленных наборами единиц, разделенных символом «+»). Алфавит: $\{+,1\}$.

№	Показатели сформированности компетенции	ФОС итогового контроля (задачи по дисциплине)
1.	В1(ПК-3)	1-12
2.	В2(ПК-3).	1-12
3.	В3(ПК-3).	1-12
4.	В4(ПК-3).	1-12
5.	В1(ПК-3).	1-12
6.	В2(ПК-3).	1-12
7.	В3(ПК-3).	1-12
8.	В4(ПК-3).	1-12

7.2 ФОС для промежуточной аттестации

7.2.1 Задания для оценки знаний

Вопросы к экзамену (ПК-3)

1. Понятие алгоритма.
2. Основные свойства алгоритмов.
3. Классификация алгоритмов.
4. Формы записи алгоритмов.
5. Понятие исполнителя алгоритмов.
6. Понятия алфавита, слова, области применимости алгоритма.
7. Процесс кодирования.
8. Блок-схемы описания алгоритмов.
9. Основные алгоритмические структуры.
10. Машина Тьюринга.
11. Тезис Тьюринга.
12. Конструирование машины Тьюринга.
13. Машина Поста.
14. Тезис Поста.
15. Конструирование машины Поста.
16. Марковские подстановки.
17. Нормальные алгоритмы Маркова.
18. Алгоритмически неразрешимые задачи.
19. Теорема Чёрча. Метод сведения.
20. Вычислимые функции.
21. Происхождение рекурсивных функций.
22. Операторы суперпозиции, примитивной рекурсии и минимизации.
23. Примитивно рекурсивные функции.
24. Частично рекурсивные и общерекурсивные функции.
25. Тезис Чёрча.
26. Понятие сложности алгоритма.
27. Временная сложность алгоритма.
28. Теоретическая сложность алгоритма.
29. Анализ алгоритма последовательного поиска в неупорядоченном массиве.
30. Анализ алгоритмов поиска минимального (максимального) элемента в неупорядоченном массиве.
31. Анализ алгоритма бинарного поиска в упорядоченном массиве.
32. Анализ алгоритма сортировки методом «пузырька».
33. Анализ алгоритма сортировки выбором.
34. Анализ алгоритма сортировки слиянием.
35. Эквивалентность различных теорий алгоритмов.
36. Машины Тьюринга и современные электронно-вычислительные машины.

№	Показатели сформированности компетенции	ФОС промежуточного контроля (вопросы к экзамену)
1.	31(ПК-3).	1-36
2.	32(ПК-3).	1-36
3.	33(ПК-3).	6-10
4.	34(ПК-3).	1-36
5.	31(ПК-3).	1-36
6.	32(ПК-3).	11-36
7.	33(ПК-3).	1-36
8.	34(ПК-3).	1-36

7.2.2. Задания для оценки умений

В качестве фондов оценочных средств для оценки умений обучающегося

используются задания, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2)

7.2.3. Задания для оценки навыков, владений, опыта деятельности

В качестве фондов оценочных средств для оценки навыков, владений, опыта деятельности обучающегося используются задания, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.3).

8. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Литература

а) Основная

1. Макоха А.Н. Математическая логика и теория алгоритмов [Электронный ресурс] : учебное пособие / А.Н. Макоха, А.В. Шапошников, В.В. Бережной. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2017. — 418 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/69397.html>

2. Брыкалова А.А. Теория алгоритмов [Электронный ресурс] : учебное пособие / А.А. Брыкалова. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 129 с. — 2227-8397.

б) Дополнительная

1. Теория алгоритмов [Электронный ресурс] : учебно-методический комплекс / . — Электрон. текстовые данные. — Алматы: Нур-Принт, 2012. — 79 с. — 9965-756-08-2.

2. Математическая логика и теория алгоритмов [Электронный ресурс] : методические указания к самостоятельной работе / . — Электрон. текстовые данные. — Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2014. — 25 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/55106.html>

3. Безусова Т.А. Теория алгоритмов. Основные подходы к формализации алгоритма [Электронный ресурс] : учебное пособие / Т.А. Безусова. — Электрон. текстовые данные. — Соликамск: Соликамский государственный педагогический институт, 2011. — 63 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/47905.html>

9. ПЕРЕЧЕНЬ КОМПЛЕКТОВ ЛИЦЕНЗИОННОГО И СВОБОДНО РАСПРОСТРАНЯЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ИСПОЛЬЗУЕМОГО ПРИ ИЗУЧЕНИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

При изучении учебной дисциплины (в том числе в интерактивной форме) предполагается применение современных информационных технологий. Комплект программного обеспечения для их использования включает в себя: операционная система Microsoft Windows 7 Pro, офисный пакет программ Microsoft Office Professional Plus 2010, офисный пакет программ Microsoft Office Professional Plus 2007, антивирусная программа Dr. Web Desktop Security Suite, архиватор 7-zip, аудиопроигрыватель AIMP, просмотр изображений FastStone Image Viewer, ПО для чтения файлов формата PDF Adobe Acrobat Reader, ПО для сканирования документов NAPS2, ПО для записи видео и проведения видеотрансляций OBS Studio, ПО для удалённого администрирования Aspiа, правовой справочник Гарант Аэро, онлайн-версия КонсультантПлюс: Студент, электронно-библиотечная система IPRBooks, электронно-библиотечная система Юрайт, математические вычисления Mathcad 14 University, версия 1С для использования типовых конфигураций в учебных целях: 1С: Предприятие 8. Комплект для обучения в высших и средних учебных заведениях, моделирование бизнес-процессов CA ERwin Process Modeler 7.3, версия 1С для обучения программированию: 1С: Предприятие 8.2 Версия для обучения программированию

10. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Российская государственная публичная библиотека <http://elibrary.rsl.ru/>
2. Информационно-правовой портал «Гарант» www.garant.ru
3. Информационно-правовой портал «КонсультантПлюс» www.consultant.ru
4. Российская государственная публичная библиотека <http://elibrary.rsl.ru/>
5. Электронно-библиотечная система (ЭБС), Издательство Юстицинформ// <http://e.lanbook.com/books/>
6. ЭБС IPRbooks (АйПиАрбукс) <http://www.iprbookshop.ru>
7. Образовательная платформа ЮРАЙТ <https://urait.ru>

11. ОБУЧЕНИЕ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Изучение данной учебной дисциплины обучающимися с ограниченными возможностями здоровья осуществляется в соответствии с Приказом Министерства образования и науки РФ от 9 ноября 2015 г. № 1309 «Об утверждении Порядка обеспечения условий доступности для инвалидов объектов и предоставляемых услуг в сфере образования, а также оказания им при этом необходимой помощи», «Методическими рекомендациями по организации образовательного процесса для инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащённости образовательного процесса» Министерства образования и науки РФ от 08.04.2014г. № АК-44/05вн, «Положением о порядке обучения студентов – инвалидов и лиц с ограниченными возможностями здоровья», утвержденным приказом ректора от 6 ноября 2015 года №60/о, «Положением о службе инклюзивного образования и психологической помощи» АНО ВО «Российский новый университет» от 20 мая 2016 года № 187/о.

Предоставление специальных технических средств обучения коллективного и индивидуального пользования, подбор и разработка учебных материалов для обучающихся с ограниченными возможностями здоровья производится преподавателями с учетом их индивидуальных психофизиологических особенностей и специфики приема передачи учебной информации.

С обучающимися по индивидуальному плану и индивидуальному графику проводятся индивидуальные занятия и консультации.

12. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля успеваемости и промежуточной аттестации

Ауд.305 (компьютерный класс № 3)

Специализированная мебель:

- столы студенческие;
- стулья студенческие;
- стол для преподавателя;
- стул для преподавателя;
- столы компьютерные;

год начала подготовки 2018

- кресла компьютерные;
- шкаф для хранения раздаточного материала;
- доска (меловая);
- маркерная доска (переносная).

Технические средства обучения:

- проектор;
- ПК для преподавателя с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду вуза;
- ПК для с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду вуза;
- веб-камера;
- экран;
- колонки;
- микрофон.

Специализированное оборудование:

- наглядные пособия (плакаты)

Автор (составитель): доцент А.С. Лабузов



(подпись)

Аннотация рабочей программы учебной дисциплины

ТЕОРИЯ АЛГОРИТМОВ

Код и направление подготовки: **09.03.03 Прикладная информатика**

Направленность (профиль): **«Прикладная информатика в экономике»**

Цели дисциплины

Обеспечение профессионального образования, способствующего социальной, академической мобильности, востребованности на рынке труда, успешной карьере, сотрудничеству.

Формирование у обучающихся систематизированных профессионально значимых знаний по теории алгоритмов и профессиональных умений и навыков, необходимых бакалавру прикладной информатики в экономике.

Изучение учебной дисциплины направлено на получение общих сведений о предмете теория алгоритмов и приобретение теоретических знаний по основным понятиям и методами теории алгоритмов с ориентацией на их использование в практической информатике и вычислительной технике.

Место дисциплины в структуре ОП бакалавриата.

Учебная дисциплина Теория алгоритмов относится к вариативной части учебного плана (Б1.В.05).

Учебная дисциплина содержательно и логически связана с другими учебными дисциплинами, изучаемыми студентами:

-предшествует освоению данной дисциплины: Математика, Информационные системы и технологии;

-после изучения данной дисциплины изучается: Математическое и имитационное моделирование, Интеллектуальные информационные системы, Исследование операций и методы оптимизации, Управление информационными системами.

Требования к уровню освоения содержания курса:

В результате освоения дисциплины обучающийся должен овладеть следующими компетенциями:

ПК-3 - Способность проектировать ИС в соответствии с профилем подготовки по видам обеспечения.

Содержание учебной дисциплины.

Тема 1. Представление об алгоритмах. Алгоритмы в математике.

1. Интуитивное представление об алгоритмах.

Интуитивное представление об алгоритмах. Неформальное понятие алгоритма. Свойства алгоритмов.

2. Алгоритмы в математике.

Способы записи алгоритмов. Основные алгоритмические структуры.

Тема 2. Машины Тьюринга и Поста. Нормальный алгоритм Маркова.

1. Машины Тьюринга и Поста.

Необходимость уточнения понятия алгоритма. Формальное описание алгоритма. Определение машины Тьюринга. Конструирование машин Тьюринга. Машина Поста. Программы для машины Поста.

2. Нормальный алгоритм Маркова.

Марковские подстановки. Нормальные алгоритмы Маркова.

Тема 3. Алгоритмически неразрешимые проблемы и вычислимые функции.

Десятая проблема Гильберта. Проблема останова. Невозможность создания универсального алгоритма отладки программы. Теорема Чёрча. Определение вычислимой функции.

Тема 4. Рекурсивные функции.

Происхождение рекурсивных функций. Операторы суперпозиции, примитивной рекурсии и минимизации. Примитивно рекурсивные функции. Частично рекурсивные и общерекурсивные функции. Тезис Чёрча.

Тема 5. Понятие сложности алгоритма.

Проблема выбора алгоритма. Определение вычислительного процесса. Временная сложность алгоритма. Теоретическая сложность алгоритма. Эффективность алгоритма.

Тема 6. Анализ алгоритмов поиска и сортировки.

Последовательный поиск в неупорядоченном массиве. Алгоритм бинарного поиска в упорядоченном массиве. Определение сортировки. «Пузырьковый» метод сортировки. Сортировка выбором. Сортировка вставками. Сортировка слиянием.

**Лист внесения изменений в рабочую программу учебной дисциплины
«Теория алгоритмов»**

Рабочая программа рассмотрена и одобрена на 2020/2021 учебный год.
Протокол № 1 заседания кафедры ПЭ от «03» сентября 2020 г.

1. Актуализация перечня основной и дополнительной учебной литературы, необходимой для освоения дисциплины на 2020-2021 учебный год.

1.1. Пункт 8.1. Основная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/432018>
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для вузов / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 117 с. — (Высшее образование). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454121>


1.2. Пункт 8.2.Дополнительная литература

1. Теория алгоритмов [Электронный ресурс] : учебно-методический комплекс / . — Электрон. текстовые данные. — Алматы: Нур-Принт, 2012. — 79 с. — 9965-756-08-2. — Режим доступа: <http://www.iprbookshop.ru/67154.html>
2. Математическая логика и теория алгоритмов [Электронный ресурс] : методические указания к самостоятельной работе / . — Электрон. текстовые данные. — Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2014. — 25 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/55106.html>

год начала подготовки 2018

3. Безусова Т.А. Теория алгоритмов. Основные подходы к формализации алгоритма [Электронный ресурс] : учебное пособие / Т.А. Безусова. — Электрон. текстовые данные. — Соликамск: Соликамский государственный педагогический институт, 2011. — 63 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/47905.html>

Зав. кафедрой

_____/Преснякова Д.В./