

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ**

УТВЕРЖДАЮ

Директор/Декан
института экономики, финансов и
управления в АПК
Гунько Юлия Александровна

«__» _____ 20__ г.

Рабочая программа дисциплины

Б1.О.19 Технологии программирования

38.03.05 Бизнес-информатика

Электронный бизнес

бакалавр

очная

1. Цель дисциплины

Целью освоения дисциплины «Технологии программирования» является изучение методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию как языков программирования, так и методов программирования.

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

Процесс изучения дисциплины направлен на формирование следующих компетенций ОП ВО и овладение следующими результатами обучения по дисциплине:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения по дисциплине
ОПК-3 Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации;	ОПК-3.2 Разрабатывает алгоритмы и программы для практического применения в сфере ИКТ	знает алгоритмы и программы для практического применения в сфере ИКТ умеет разрабатывать алгоритмы и программы для практического применения в сфере ИКТ владеет навыками навыками разработки алгоритмов и программ для практического применения в сфере ИКТ
ОПК-6 Способен выполнять отдельные задачи в рамках коллективной научно-исследовательской, проектной и учебно-профессиональной деятельности для поиска, выработки и применения новых решений в области информационно-коммуникационных технологий.	ОПК-6.4 Решает профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования	знает естественнонаучные и общеинженерные знания, методы математического анализа и моделирования умеет Решать профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования Знать владеет навыками навыками решения профессиональных задач с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования Знать
ПК-1 Управление информационными ресурсами	ПК-1.2 Управление информацией из различных источников	знает принципы работы с источниками данных, методы их верификации и систематизации информации умеет осуществлять сбор, фильтрацию и структурирование информации из разнородных источников владеет навыками технологиями анализа данных, инструментами мониторинга и управления информационными потоками

3. Место дисциплины в структуре образовательной программы

Дисциплина «Технологии программирования» является дисциплиной обязательной части программы.

Изучение дисциплины осуществляется в 2семестре(-ах).

Для освоения дисциплины «Технологии программирования» студенты используют знания, умения и навыки, сформированные в процессе изучения дисциплин:

Операционные системы

Высшая математика

Введение в профессиональную деятельность

Ознакомительная практикаОперационные системы

Операционные системы

Высшая математика

Введение в профессиональную деятельность

Ознакомительная практикаВысшая математика

Операционные системы

Высшая математика

Введение в профессиональную деятельность

Ознакомительная практикаВведение в профессиональную деятельность

Операционные системы

Высшая математика

Введение в профессиональную деятельность

Ознакомительная практикаОзнакомительная практика

Освоение дисциплины «Технологии программирования» является необходимой основой для последующего изучения следующих дисциплин:

Цифровые технологии в профессиональной деятельности

Подготовка к сдаче и сдача государственного экзамена

Выполнение и защита выпускной квалификационной работы

Технологическая (проектно-технологическая) практика

Технологическая (проектно-технологическая) практика

Интернет-маркетинг

Информационные технологии электронного бизнеса

Программирование в бизнес-системах

Электронная коммерция

Управление IT проектами

Базы данных

Алгоритмы и структуры данных

Сетевые технологии

Web-программирование

Управление процессами жизненного цикла ИС

Логистика

Конфигурирование в информационных системах

ERP-системы

Системы искусственного интеллекта

Методы оптимальных решений

Преддипломная практика

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

Общая трудоемкость дисциплины «Технологии программирования» в соответствии с рабочим учебным планом и ее распределение по видам работ представлены ниже.

Семестр	Трудоемк	Контактная работа с преподавателем, час	Самостоя-	Контроль,	Форма
---------	----------	---	-----------	-----------	-------

	ость час/з.е.	лек- ции	практические занятия	лабораторные занятия	тельная ра- бота, час	час	промежуточной аттестации (форма контроля)
2	180/5	18		54	72	36	Эк
в т.ч. часов: в интерактивной форме		4		12			
практической подготовки		10		26	36		

Семестр	Трудоёмк ость час/з.е.	Внеаудиторная контактная работа с преподавателем, час/чел					
		Курсовая работа	Курсовой проект	Зачет	Дифференцирован ный зачет	Консультации перед экзаменом	Экзамен
2	180/5						0.25

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

№	Наименование раздела/темы	Семестр	Количество часов					Формы текущего контроля успеваемости и промежуточной аттестации	Оценочное средство проверки результатов достижения индикаторов компетенций	Код индикат оров достиж ения компете нций
			всего	Лекции	Семинарск ие занятия		Самостоятельная работа			
					Практические	Лабораторные				
1.	1 раздел. 1									
1.1.	Введение в алгоритмизацию и программирование	2	10	6		4	4	Тест	ОПК-3.2, ОПК-6.4, ПК-1.2	
1.2.	Структурный подход к программированию	2	18	8		10	16	Тест	ОПК-3.2, ОПК-6.4, ПК-1.2	
1.3.	Кт 1	2	2			2		КТ 1	ОПК-3.2, ОПК-6.4, ПК-1.2	
1.4.	Модульное программирование. Программирование абстрактных типов данных	2	12	2		10	22	Тест	ОПК-3.2, ОПК-6.4, ПК-1.2	
1.5.	Объектно-ориентированное программирование	2	28	2		26	30	Тест	ОПК-3.2, ОПК-6.4, ПК-1.2	

1.6.	Кт 2	2	2			2		КТ 2	Тест	ОПК-3.2, ОПК-6.4, ПК-1.2
	Промежуточная аттестация	Эк								
	Итого		180	18		54	72			
	Итого		180	18		54	72			

5.1. Лекционный курс с указанием видов интерактивной формы проведения занятий

Тема лекции (и/или наименование раздел) (вид интерактивной формы проведения занятий)/ (практическая подготовка)	Содержание темы (и/или раздела)	Всего, часов / часов интерактивных занятий/ практическая подготовка
Введение в алгоритмизацию и программирование	Методологии программирования	2/2
Введение в алгоритмизацию и программирование	Алгоритмические структуры	2/2
Введение в алгоритмизацию и программирование	Синтаксис и семантика формального языка	2/2
Структурный подход к программированию	Основные конструкции алгоритмических языков	2/-
Структурный подход к программированию	Основные операторы языка	2/-
Структурный подход к программированию	Простые типы языка программирования	2/2
Структурный подход к программированию	Структурированные типы языка программирования высокого уровня	1/-
Структурный подход к программированию	Алгоритмы поиска и сортировки	1/-
Модульное программирование. Программирование абстрактных типов данных	Процедуры и функции	2/-
Объектно-ориентированное программирование	Введение в объектно-ориентированное программирование	2/-
Итого		18

5.2.2. Лабораторные занятия с указанием видов проведения занятий в интерактивной форме

Наименование раздела дисциплины	Формы проведения и темы занятий (вид интерактивной формы проведения занятий)/(практическая подготовка)	Всего, часов / часов интерактивных занятий/ практическая подготовка	
		вид	часы
Введение в алгоритмизацию и	Алгоритмические структуры	лаб.	2

программирование			
Введение в алгоритмизацию и программирование	Контрольная точка №1	лаб.	2
Структурный подход к программированию	Основные операторы языка	лаб.	2
Структурный подход к программированию	Структурированные типы языка программирования высокого уровня	лаб.	6
Структурный подход к программированию	Контрольная точка №2	лаб.	2
Кт 1	Кт 1	лаб.	2
Модульное программирование. Программирование абстрактных типов данных	Процедуры и функции	лаб.	4
Модульное программирование. Программирование абстрактных типов данных	Организация динамических структур данных	лаб.	4
Модульное программирование. Программирование абстрактных типов данных	Контрольная точка №3	лаб.	2
Объектно-ориентированное программирование	Введение в объектно-ориентированное программирование	лаб.	12
Объектно-ориентированное программирование	Реализация абстракций данных методами объектно-ориентированного программирования	лаб.	12
Объектно-ориентированное программирование	Контрольная точка №4	лаб.	2
Кт 2	Кт 2	лаб.	2

5.3. Курсовой проект (работа) учебным планом не предусмотрен

5.4. Самостоятельная работа обучающегося

Темы и/или виды самостоятельной работы	Часы
Изучение пройденного материала и подготовка к лабораторной работе	2

Изучение пройденного материала и подготовка к контрольной точке	2
Изучение пройденного материала и подготовка к лабораторной работе	2
Изучение пройденного материала и подготовка к лабораторной работе	2
Изучение пройденного материала и подготовка к лабораторной работе	6
Изучение пройденного материала и подготовка к контрольной точке	6
Изучение пройденного материала и подготовка к лабораторной работе	6
Изучение пройденного материала и подготовка к лабораторной работе	8
Изучение пройденного материала и подготовка к контрольной точке	8
Изучение пройденного материала и подготовка к лабораторной работе	10
Изучение пройденного материала и подготовка к лабораторной работе	8

Изучение пройденного материала и подготовка к контрольной точке	12
---	----

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Учебно-методическое обеспечение для самостоятельной работы обучающегося по дисциплине «Технологии программирования» размещено в электронной информационно-образовательной среде Университета и доступно для обучающегося через его личный кабинет на сайте Университета. Учебно-методическое обеспечение включает:

1. Рабочую программу дисциплины «Технологии программирования».
2. Методические рекомендации для организации самостоятельной работы обучающегося по дисциплине «Технологии программирования».
3. Методические рекомендации по выполнению письменных работ () (при наличии).
4. Методические рекомендации по выполнению контрольной работы студентами заочной формы обучения (при наличии)
5. Методические указания по выполнению курсовой работы (проекта) (при наличии).

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации:

№ п/п	Темы для самостоятельного изучения	Рекомендуемые источники информации (№ источника)		
		основная (из п.8 РПД)	дополнительная (из п.8 РПД)	метод. лит. (из п.8 РПД)
1	Введение в алгоритмизацию и программирование. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
2	Введение в алгоритмизацию и программирование. Изучение пройденного материала и подготовка к контрольной точке	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
3	Структурный подход к программированию. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
4	Структурный подход к программированию. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
5	Структурный подход к программированию. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
6	Структурный подход к программированию. Изучение пройденного материала и подготовка к контрольной точке	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
7	Модульное программирование. Программирование абстрактных типов данных. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
8	Модульное программирование. Программирование абстрактных типов данных. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2

9	Модульное программирование. Программирование абстрактных типов данных. Изучение пройденного материала и подготовка к контрольной точке	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
10	Объектно-ориентированное программирование. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
11	Объектно-ориентированное программирование. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2
12	Объектно-ориентированное программирование. Изучение пройденного материала и подготовка к контрольной точке	Л1.1, Л1.2, Л1.3, Л1.4	Л2.1, Л2.2, Л2.3, Л2.4, Л2.5	Л3.1, Л3.2

7. Фонд оценочных средств (оценочных материалов) для проведения промежуточной аттестации обучающихся по дисциплине «Технологии программирования»

7.1. Перечень индикаторов компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Индикатор компетенции (код и содержание)	Дисциплины/элементы программы (практики, ГИА), участвующие в формировании индикатора компетенции	1		2		3		4	
		1	2	3	4	5	6	7	8
ОПК-3.2: Разрабатывает алгоритмы и программы для практического применения в сфере ИКТ	Алгоритмы и структуры данных			x					
	Базы данных				x				
	Ознакомительная практика		x						
	Программирование в бизнес-системах			x					
	Технологическая (проектно-технологическая) практика				x		x		
ОПК-6.4: Решает профессиональные задачи с применением естественнонаучных и инженерных знаний, методов математического анализа и моделирования	Алгоритмы и структуры данных			x					
	Высшая математика	x	x						
	Методы оптимальных решений				x				
	Ознакомительная практика		x						
	Расчетно-аналитические методы и модели	x	x	x	x	x	x	x	
Технологическая (проектно-технологическая) практика				x		x			
ПК-1.2: Управление информацией из различных источников	ERP-системы							x	
	Конфигурирование в информационных системах						x		
	Операционные системы	x							
	Преддипломная практика								x
	Программирование в бизнес-системах			x					
Системы искусственного интеллекта							x		

7.2. Критерии и шкалы оценивания уровня усвоения индикатора компетенций, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Оценка знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций по дисциплине «Технологии программирования» проводится 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 её корректировке, а также для совершенствования методики обучения, организации учебной работы и оказания индивидуальной помощи обучающемуся.

Промежуточная аттестация по дисциплине «Технологии программирования» проводится в виде Экзамен.

За знания, умения и навыки, приобретенные студентами в период их обучения, выставляются оценки «ЗАЧТЕНО», «НЕ ЗАЧТЕНО». (или «ОТЛИЧНО», «ХОРОШО», «УДОВЛЕТВОРИТЕЛЬНО», «НЕУДОВЛЕТВОРИТЕЛЬНО» для дифференцированного зачета/экзамена)

Для оценивания знаний, умений, навыков и (или) опыта деятельности в университете применяется балльно-рейтинговая система оценки качества освоения образовательной программы. Оценка проводится при проведении текущего контроля успеваемости и промежуточных аттестаций обучающихся. Рейтинговая оценка знаний является интегрированным показателем качества теоретических и практических знаний и навыков студентов по дисциплине.

Состав балльно-рейтинговой оценки студентов очной формы обучения

Для студентов очной формы обучения знания по осваиваемым компетенциям формируются на лекционных и практических занятиях, а также в процессе самостоятельной подготовки.

В соответствии с балльно-рейтинговой системой оценки, принятой в Университете студентам начисляются баллы по следующим видам работ:

№ контрольной точки	Оценочное средство результатов индикаторов достижения компетенций		Максимальное количество баллов
2 семестр			
КТ 1	Тест		15
КТ 2	Тест		15
Сумма баллов по итогам текущего контроля			30
Посещение лекционных занятий			20
Посещение практических/лабораторных занятий			20
Результативность работы на практических/лабораторных занятиях			30
Итого			100
№ контрольной точки	Оценочное средство результатов индикаторов достижений компетенций	Максимальное количество баллов	Критерии оценки знаний студентов
2 семестр			

КТ 1	Тест	15	11-15 баллов выставляется обучающемуся, если тестовые задания выполняются на 85% и выше; 8-10 баллов выставляется обучающемуся, если тестовые задания выполняются на 70 - 84%; 5-7 баллов выставляется обучающемуся, если тестовые задания выполняются на 55 – 69 %; 1-4 балла выставляется обучающемуся, если тестовые задания выполняются на 45 – 54%; 0 баллов выставляется обучающемуся, если тестовые задания выполняются на 44% и меньше.
КТ 2	Тест	15	11-15 баллов выставляется обучающемуся, если тестовые задания выполняются на 85% и выше; 8-10 баллов выставляется обучающемуся, если тестовые задания выполняются на 70 - 84%; 5-7 баллов выставляется обучающемуся, если тестовые задания выполняются на 55 – 69 %; 1-4 балла выставляется обучающемуся, если тестовые задания выполняются на 45 – 54%; 0 баллов выставляется обучающемуся, если тестовые задания выполняются на 44% и меньше.

Критерии и шкалы оценивания результатов обучения на промежуточной аттестации

При проведении итоговой аттестации «зачет» («дифференцированный зачет», «экзамен») преподавателю с согласия студента разрешается выставлять оценки («отлично», «хорошо», «удовлетворительно», «зачет») по результатам набранных баллов в ходе текущего контроля успеваемости в семестре по выше приведенной шкале.

В случае отказа – студент сдает зачет (дифференцированный зачет, экзамен) по приведенным выше вопросам и заданиям. Итоговая успеваемость (зачет, дифференцированный зачет, экзамен) не может оцениваться ниже суммы баллов, которую студент набрал по итогам текущей и промежуточной успеваемости.

При сдаче (зачета, дифференцированного зачета, экзамена) к заработанным в течение семестра студентом баллам прибавляются баллы, полученные на (зачете, дифференцированном зачете, экзамене) и сумма баллов переводится в оценку.

Критерии и шкалы оценивания ответа на экзамене

Сдача экзамена может добавить к текущей балльно-рейтинговой оценке студентов не более 20 баллов:

Содержание билета	Количество баллов
Теоретический вопрос №1	до 7
Теоретический вопрос №2	до 7
Задача (оценка умений и	до 6
Итого	20

Критерии оценки ответа на экзамене

Теоретические вопросы (вопрос 1, вопрос 2)

7 баллов выставляется студенту, полностью освоившему материал дисциплины или курса в соответствии с учебной программой, включая вопросы рассматриваемые в рекомендованной программой дополнительной справочно-нормативной и научно-технической литературы, свободно владеющему основными понятиями дисциплины. Требуется полное понимание и четкость изложения ответов по экзаменационному заданию (билету) и дополнительным вопросам, заданных экзаменатором. Дополнительные вопросы, как правило, должны относиться к материалу дисциплины или курса, не отраженному в основном экзаменационном задании (билете) и выявляют полноту знаний студента по дисциплине.

5 балла заслуживает студент, ответивший полностью и без ошибок на вопросы экзаменационного задания и показавший знания основных понятий дисциплины в соответствии с обязательной программой курса и рекомендованной основной литературой.

3 балла дан недостаточно полный и недостаточно развернутый ответ. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. Студент не способен самостоятельно выделить существенные и несущественные признаки и причинно-следственные связи. Студент может конкретизировать обобщенные знания, доказав на примерах их основные положения только с помощью преподавателя. Речевое оформление требует поправок, коррекции.

2 балла дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствуют фрагментарность, нелогичность изложения. Студент не осознает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не только на поставленный вопрос, но и на другие вопросы дисциплины.

1 балл дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствуют фрагментарность, нелогичность изложения. Студент не осознает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не только на поставленный вопрос, но и на другие вопросы дисциплины.

0 баллов - при полном отсутствии ответа, имеющего отношение к вопросу.

Оценивание задачи

6 баллов Задачи решены в полном объеме с соблюдением необходимой последовательности.

5 баллов

4 балла Задачи решены с небольшими недочетами.

3 балла

2 балла Задачи решены не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы.

1 баллов Задачи решены частично, с большим количеством вычислительных ошибок, объем выполненной части работы не позволяет сделать правильных выводов.

0 баллов Задачи не решены или работа выполнена не полностью, и объем выполненной части работы не позволяет сделать правильных выводов.

Перевод рейтинговых баллов в пятибалльную систему оценки знаний обучающихся: для экзамена:

- «отлично» – от 89 до 100 баллов – теоретическое содержание курса освоено полностью, без пробелов необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному;

- «хорошо» – от 77 до 88 баллов – теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками;

- «удовлетворительно» – от 65 до 76 баллов – теоретическое содержание курса освоено ча-

стично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки;

- «неудовлетворительно» – от 0 до 64 баллов - теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к существенному повышению качества выполнения учебных заданий

7.3. Примерные оценочные материалы для текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины «Технологии программирования»

1. Системы программирования: основные понятия и определения
2. Состав системы программирования
3. Языки программирования: основные понятия и определения
4. Уровни языков программирования
5. Классификация языков программирования
6. Основные понятия кодирования информации и системы счисления
7. Виды систем счисления
8. Двоичная системы счисления
9. Десятичная системы счисления
10. Восьмеричная система счисления
11. Шестнадцатеричная система счисления
12. Порядок решения задачи алгоритмизации
13. Этапы решения задач на ПК
14. Алгоритм и его свойства
15. Способы описания алгоритмов
16. Словесно-формульный способ описания алгоритмов
17. Запись алгоритма на псевдокоде
18. Программный способ описания алгоритмов
19. Графический способ описания алгоритмов
20. Виды алгоритмов и принципы их составления
21. Характеристика линейных и разветвляющихся алгоритмов.
22. Алгоритм циклических вычислительных процессов, вложенные циклы.
23. Алгоритм комбинированных вычислительных процессов
24. Понятия и типы массивов
25. Алгоритмизация задач по обработке одномерных массивов
26. Алгоритмизация задач по обработке многомерных массивов
27. Язык программирования Python: область использования
28. Язык программирования Python: назначения
29. Язык программирования Python: основные проекты
30. История создания языка Python
31. Версии языка Python
32. Возможности языка Python и его предпосылки применения в обучении
33. Преимущества и недостатки Python
34. Установка и работа в Python
35. Синтаксис языка Python
36. Алфавит языка Python
37. Алфавит и словарь языка Python
38. Типы данных, используемых в Python
39. Выражения и операции в Python
40. Операторы присваивания, ввода, вывода, комментариев в Python
41. Стандартные функции в Python
42. Разработка на Python: плюсы и минусы
43. Примеры использования Python крупными компаниями
44. Достоинства и недостатки Python для разработки

45. Трудоустройство и средняя зарплата Python-разработчика
46. Должности и требования к Python-разработчику
47. Интегрированные среды для Python
48. Сравнение Python с другими языками программирования
49. Популярные языки программирования в 2024 году
50. Операторы ветвления в Python: формы, составные условия, вложенные ветвления
51. Операторы цикла в Python: цикл с предусловием (пример)
52. Операторы цикла в Python: цикл с постусловием (пример)
53. Операторы цикла в Python: цикл с параметром (пример)
54. Массивы в Python: понятие, виды, описание
55. Вывод массива на экран в Python
56. Способы заполнения массивов в Python
57. Сортировка массива в Python
58. Строковые величины в языке Python
59. Процедуры в языке Python
60. Функции в языке Python

Примерные практические задания на экзамене

Язык программирования Python

1. Вычислить и вывести на экран монитора в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от $X_{нач}$ до $X_{кон}$ с шагом dx и точностью ϵ . Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда

2. Даны две матрицы одного порядка $m * n$ (m строк x n столбцов).

Написать программу сложения, вычитания и транспонирования матриц.

1. Сложение и вычитание:

2. Транспонирование

3. Написать программу на Python, которая по введённому значению аргумента вычисляет значение функции, заданной в виде графика.

Гиперболический косинус может быть описан формулой:

4. Вывести на экран средствами пакета Tkinter в графическом режиме графики двух функций на интервале от $X_{нач}$ до $X_{кон}$ с шагом dx . Первая функция задана с помощью ряда Тейлора, ее вычисление должно выполняться с точностью ϵ . Значение параметра b для второй функции вводится с клавиатуры. Графики должны быть плавными и различаться цветами.

5. Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

Тематика рефератов

1. История языков программирования.
2. Язык компьютера и человека.
3. Объектно-ориентированное программирование.
4. Непроцедурные системы программирования.
5. Искусственный интеллект и логическое программирование.
6. Языки манипулирования данными в реляционных моделях.
7. Макропрограммирование в среде Microsoft OFFICE.
8. «Визуальное» программирование. VISUAL BASIC, C, PROLOG.
9. Все о DELPHI.
10. Программирование на HTML, JAVA.
11. Издательская система TeX как система программирования.
12. Современные парадигмы программирования. Что дальше?
13. Никлаус Вирт. Структурное программирование. Pascal и Modula.
14. Что мы знаем о Fortran?
15. История языка Бейсик.

16. Язык Ассемблера.
17. Алгоритмический язык Ершова.
18. Все о Logo-мирах.
19. История программирования в лицах.
20. Язык программирования ADA.
21. Язык программирования PL/1.
22. Язык программирования Algol.
23. Язык программирования Си.
24. О фирмах-разработчиках систем программирования.
25. Языки программирования в СУБД.
26. О системах программирования для учебных целей.

Тематика курсовой работы

1. Значение выбора языка, платформы и технологии программирования.
2. Важность структурирования программ и использования принципа модульности.
3. Варианты использования автокодов для оптимизации процессов.
4. Виды диаграмм и их значение в программировании.
5. Влияние программистов на эффективность организации рабочего процесса предприятий.
6. Значение процедурного программирования в 21 веке.
7. Конструкция императивных языков.
8. Макросы в ассемблере.
9. Оптимизация программного обеспечения для успешной и бесперебойной работы предприятия.
10. Особенности структурного программирования.
11. Проблемы современного программирования и их решение.
12. Программирование графических приложений с использованием графического интерфейса устройств (GDI+).
13. Особенности многоязыкового программирования.
14. Программирование как неотъемлемая часть развития общества.
15. Значение выбора языка, платформы и технологии программирования.
16. Программирование приложений с использованием библиотеки классов .NET Framework.
17. Программирование приложений, поддерживающих многодокументный интерфейс (MDI).
18. Программирование приложений, поддерживающих однодокументный интерфейс (SDI).
19. Программирование приложений, построенных на основе архитектуры «Документ/Представление».
20. Программирование с информационным сервером INTERNET (ISAPI).
21. Программирование сетевых приложений с использованием функций API.
22. Практическое использование генетического программирования.
23. Расширение областей программирования как следствие цифровизации современного общества.
24. Создание и индексация массивов.
25. Способы разработки мобильных приложений.
26. Сфера применения генерации случайных чисел.
27. Типизация в языках программирования.
28. Функции. Их параметры и аргументы в современном программировании.
29. Интерфейс во взаимодействии с другими системами программирования.
30. Язык СИ. Достоинства, недостатки и варианты применения.

ЗНАТЬ

Задание 1

Стартап разрабатывает новый мобильный банковский продукт в условиях высокой

неопределённости требований и жёстких сроков выхода на рынок. Какой подход к управлению проектом будет наиболее предпочтителен?

Варианты:

- A) Каскадная модель (Waterfall)
- B) Гибкая методология (Agile, например, Scrum)
- C) V-образная модель
- D) RUP (Rational Unified Process)

Ответ: B

Задание 2

В бизнес-логике расчёта скидок для интернет-магазина один и тот же алгоритм проверки статуса постоянного клиента используется в 12 разных местах кода. Что это нарушает и как лучше исправить?

Варианты:

A) Это нарушение принципа DRY. Нужно вынести алгоритм в отдельную функцию/метод и вызывать её.

B) Это нормальная практика для ускорения работы программы.

C) Это нарушение принципа KISS. Нужно удалить лишние проверки.

D) Ничего исправлять не нужно, если код работает.

Ответ: A

Задание 3

Команда из 5 человек одновременно работает над разными функциями веб-приложения для CRM-системы. Какой инструмент критически необходим для слияния их изменений, отслеживания истории правок и отката ошибок?

Варианты:

A) Система управления базами данных (СУБД)

B) Интегрированная среда разработки (IDE)

C) Система контроля версий (Git)

D) Система баг-трекинга (Jira)

Ответ: C

Задание 4

После завершения разработки модуля «Формирование платёжной ведомости» необходимо проверить, корректно ли он взаимодействует с модулем «Налоговый учёт». Какой тип тестирования описывается?

Варианты:

A) Модульное (Unit) тестирование

B) Интеграционное (Integration) тестирование

C) Приёмочное (UAT) тестирование

D) Нагрузочное (Load) тестирование

Ответ: B

Задание 5

На каком этапе жизненного цикла программного обеспечения определяются ключевые бизнес-требования, ограничения бюджета и сроки?

Варианты:

A) Проектирование (Design)

B) Реализация (Implementation)

C) Сопровождение (Maintenance)

D) Сбор требований (Requirements Gathering & Analysis)

Ответ: D

Задание 6

В системе управления складом есть различные алгоритмы расчёта стоимости доставки (для курьера, почты, самовывоза). Требуется организовать код так, чтобы можно было легко добавлять новые алгоритмы, не изменяя основной код заказа. Какой паттерн проектирования лучше применить?

Варианты:

A) Фабричный метод (Factory Method)

B) Одиночка (Singleton)

- C) Стратегия (Strategy)
- D) Наблюдатель (Observer)

Ответ: C

Задание 7

Чтобы успеть к запуску промо-акции, команда разработки внедрила функцию быстрой оплаты, используя упрощённые, но неоптимальные решения. Что образуется в коде и проекте вследствие такого решения?

Варианты:

- A) Накопленный капитал
- B) Технический долг
- C) Бизнес-актив
- D) Резервная копия

Ответ: B

Задание 8

Для высоконагруженного сервиса аналитики в реальном времени, который обрабатывает поток кликов с сайта, требуется хранить слабоструктурированные данные с возможностью горизонтального масштабирования. Какой тип базы данных наиболее предпочтителен?

Варианты:

- A) Реляционная СУБД (MySQL, PostgreSQL)
- B) Документоориентированная NoSQL БД (MongoDB)
- C) Графовая БД (Neo4j)
- D) SQLite

Ответ: B

Задание 9

Что является основной целью рефакторинга кода?

Варианты:

- A) Добавление новой функциональности для пользователя.
- B) Исправление критических ошибок (багов).
- C) Улучшение внутренней структуры кода без изменения его внешнего поведения.
- D) Ускорение работы программы за счёт оптимизации алгоритмов.

Ответ: C

Задание 10

Как называется практика и набор инструментов, которые позволяют автоматически собирать, тестировать и разворачивать приложение при каждом изменении кода, что ускоряет доставку обновлений пользователям?

Варианты:

- A) Система управления проектами (MS Project)
- B) Интеграционная среда разработки (Visual Studio)
- C) Непрерывная интеграция и непрерывная поставка (CI/CD)
- D) Виртуализация (Docker)

Ответ: C

Задание 11

Кто в команде разработки преимущественно отвечает за перевод требований бизнеса (заказчика) в формализованные спецификации и пользовательские сценарии (User Stories), понятные разработчикам?

Варианты:

- A) Системный администратор
- B) Тестирующий (QA Engineer)
- C) Бизнес-аналитик
- D) Тимлид (Team Lead)

Ответ: C

Задание 12

С точки зрения бизнеса, какое из перечисленных является главным экономическим преимуществом использования облачной инфраструктуры (IaaS, например, AWS, Azure) вместо содержания собственного серверного оборудования?

Варианты:

- A) Перевод капитальных затрат (CAPEX) в операционные (OPEX).
- B) Полный контроль над физическим оборудованием.
- C) Гарантированная максимальная скорость работы.
- D) Отсутствие необходимости в IT-специалистах.

Ответ: A

Задание 13

Чем микросервисная архитектура принципиально отличается от монолитной?

Варианты:

- A) Приложение разбито на множество небольших, слабосвязанных сервисов, развёртываемых независимо.
- B) Приложение написано на более современных языках программирования.
- C) Все компоненты приложения тесно связаны и развёртываются как единое целое.
- D) Она всегда работает быстрее монолита.

Ответ: A

Задание 14

Злоумышленник смог внедрить в поле ввода на сайте фрагмент SQL-кода, который изменил запрос к базе данных и позволил получить доступ к чужим данным. Какая уязвимость была использована?

Варианты:

- A) XSS (Межсайтовый скриптинг)
- B) SQL-инъекция
- C) DDoS-атака
- D) Фишинг

Ответ: B

Задание 15

В User Story для функции «Восстановление пароля» указано: «Пользователь должен получить временную ссылку для сброса пароля на email». Что из перечисленного является хорошим, проверяемым критерием приемлемости?

Варианты:

- A) Ссылка должна быть красивой.
- B) При вводе зарегистрированного email в форму восстановления пароля система отправляет письмо с уникальной ссылкой, активной в течение 1 часа, на этот адрес.
- C) Система должна быстро отправлять письма.
- D) Это важная функция для безопасности.

Ответ: B

Задание 1

Этапы классического жизненного цикла разработки ПО (Waterfall). Установите правильный порядок этапов.

1. Сопровождение
2. Проектирование
3. Тестирование
4. Анализ требований
5. Реализация (Кодирование)

Ответ: 4 → 2 → 5 → 3 → 1

Задание 2

Приоритет действий при получении задачи на исправление бага (bugfix). Расположите действия в логическом порядке.

1. Написать автоматический тест, воспроизводящий баг.
2. Проанализировать логи и стэк-трейс ошибки.
3. Воспроизвести баг на тестовом окружении.
4. Отправить исправление в репозиторий и запустить сборку.
5. Локализовать причину в исходном коде.

Ответ: 2 → 3 → 1 → 5 → 4

Задание 3

Стадии процесса рефакторинга кода. Установите последовательность.

1. Внести минимальные изменения в код.
2. Убедиться, что существующие тесты проходят.
3. Определить «запах кода» (code smell), требующий улучшения.
4. Написать тесты для покрытия рефакторимого участка (если их нет).
5. Провести рефакторинг, не меняя поведения.

Ответ: 3 → 4 → 2 → 5 → 1

Задание 4

Установите порядок создания простой пользовательской истории в формате INVEST

1. Уточняющий критерий приемки (Acceptance Criteria)
2. Роль (As a...)
3. Цель/ценность (So that...)
4. Действие (I want to...)

Ответ: 2 → 4 → 3 → 1

Задание 5

Установите правильный порядок типичного workflow.

1. Создать новую ветку (git checkout -b feature_branch)
2. Зафиксировать изменения (git commit -m "...")
3. Влить ветку в main (git merge / pull request)
4. Сделать коммиты локально
5. Получить последние изменения с главной ветки (git pull origin main)

Ответ: 5 → 1 → 4 → 2 → 3

Задание 6

Установите соответствие между артефактами гибкой методологии (Scrum) и их описанием

Артефакты:

- А. Бэклог продукта (Product Backlog)
- В. Бэклог спринта (Sprint Backlog)
- С. Инкремент продукта (Increment)

Описания:

1. Упорядоченный список всего, что может понадобиться в продукте, источник требований.
2. Рабочая версия продукта с новыми функциями, выполненных в конце спринта.
3. Набор элементов из бэклога продукта, выбранных для спринта, план реализации.

Ответ: А-1, В-3, С-2

Задание 7

Установите соответствие типов тестирования и их целей

Типы тестирования:

- А. Модульное (Unit)
- В. Интеграционное (Integration)
- С. Приемочное (Acceptance)

Цели:

1. Проверить соответствие системы бизнес-требованиям и готовность к использованию заказчиком.
2. Проверить работу отдельного класса, функции или модуля в изоляции.
3. Проверить взаимодействие между несколькими модулями, компонентами или системами.

Ответ: А-2, В-3, С-1

Задание 8

Установите соответствие «запахов кода» (code smells) и возможных методов рефакторинга

Запахи кода:

- А. Длинный метод (Long Method)
- В. Повторение кода (Duplicate Code)
- С. Большой класс (Large Class)

Методы рефакторинга:

1. Выделение метода (Extract Method), выделение класса (Extract Class)
2. Выделение метода (Extract Method), выделение подкласса (Extract Subclass)
3. Выделение метода (Extract Method), перенос метода (Move Method)

Ответ: А-1, В-3, С-2

Задание 9

Установите соответствие видов архитектурных шаблонов и их ключевых характеристик

Шаблоны:

А. Многослойная архитектура (Layered Architecture)

В. Микросервисная архитектура (Microservices)

С. Событийно-ориентированная архитектура (Event-Driven)

Характеристики:

1. Система состоит из небольших, независимо развертываемых сервисов.
2. Компоненты общаются через асинхронную отправку и обработку событий.
3. Система разделена на слои (представление, бизнес-логика, данные), каждый зависит

от нижележащего.

Ответ: А-3, В-1, С-2

Задание 10

Установите соответствие принципов SOLID и их кратких трактовок

Принципы:

А. Single Responsibility

В. Open/Closed

С. Liskov Substitution

Трактовки:

1. Классы должны быть открыты для расширения, но закрыты для модификации.
2. Дочерний класс должен быть способен заменить родительский, не ломая программу.
3. У класса должна быть только одна причина для изменения.

Ответ: А-3, В-1, С-2

Задание 11

Установите соответствие видов NoSQL баз данных и типичных сценариев использования в бизнесе

Виды БД:

А. Документная (MongoDB)

В. Колоночная (Cassandra)

С. Графовая (Neo4j)

Сценарии:

1. Хранение иерархических каталогов товаров, профилей пользователей с динамическими атрибутами.
2. Аналитика больших данных, обработка событий с высокой скоростью записи.
3. Рекомендательные системы, анализ социальных связей, обнаружение мошенничества.

Ответ: А-1, В-2, С-3

Задание 12

Установите соответствие ролей в команде разработки и их ключевых задач

Роли:

А. Бизнес-аналитик (Business Analyst, BA)

В. Разработчик (Developer)

С. Владелец продукта (Product Owner, PO)

Задачи:

1. Формирует видение продукта, расставляет приоритеты в бэклоге, максимизирует ценность.
2. Выявляет потребности стейкхолдеров, формализует требования, пишет пользовательские истории.
3. Проектирует, пишет, тестирует и отлаживает код согласно техническому заданию.

Ответ: А-2, В-3, С-1

Задание 13

Установите соответствие инструментов и их основного назначения в процессе разработки

Инструменты:

А. Jira / YouTrack

В. Jenkins / GitLab CI

C. Swagger / OpenAPI

Назначение:

1. Непрерывная интеграция и непрерывное развертывание (CI/CD).
2. Описание, создание и документирование REST API.
3. Отслеживание задач, багов и agile-планирование.

Ответ: A-3, B-1, C-2

Задание 14

Установите соответствие метрик качества ПО и их описания

Метрики:

- A. Cyclomatic Complexity
- B. Code Coverage
- C. Mean Time To Recovery (MTTR)

Описания:

1. Мера того, какая часть исходного кода выполнена при прохождении тестов.
2. Среднее время, необходимое для восстановления работоспособности системы после сбоя.
3. Количество линейно независимых путей в коде, мера его сложности.

Ответ: A-3, B-1, C-2

Задание 15

Установите соответствие паттернов проектирования (GoF) и решаемых проблем

Паттерны:

- A. Фасад (Facade)
- B. Стратегия (Strategy)
- C. Наблюдатель (Observer)

Проблемы:

1. Необходимость уведомлять множество объектов об изменении состояния другого объекта.
2. Предоставление простого интерфейса к сложной подсистеме.
3. Возможность выбирать алгоритм из семейства алгоритмов во время выполнения.

Ответ: A-2, B-3, C-1

Задание 1

Этот принцип программирования (из аббревиатуры SOLID) гласит, что класс должен быть открыт для расширения, но закрыт для модификации. Напишите его название.

Ответ: Принцип открытости

Задание 2

Как называется список всех функций, улучшений и исправлений, запланированных для продукта, который постоянно обновляется и приоритизируется? (Ключевой артефакт в Scrum).

Ответ: Бэклог продукта

Задание 3

Какое минимальное число программистов должно быть в команде, чтобы практика парного программирования была возможна?

Ответ: 2

Задание 4

Как называется процесс объединения копий рабочих версий (веток) из разных источников в единую ветку, например, в main? (Ключевое действие в Git).

Ответ: Слияние

Задание 5

Если для метода calculateDiscount(order) в качестве входных данных мы передаем заказ с отрицательной суммой, чтобы проверить, как программа обрабатывает некорректные данные, — какой это вид тестирования?

Ответ: Негативное тестирование

Задание 6

Какой шаблон проектирования позволяет объектам оповещать другие объекты об изменении своего состояния? (Часто используется в системах обработки событий).

Ответ: Наблюдатель

Задание 7

Как называется язык разметки, который используется для описания и документирования REST API в удобной для человека и машины форме? (Пример инструмента — Swagger).

Ответ: OpenAPI

Задание 8

В контексте жизненного цикла ПО, как называется этап, на котором команда исправляет найденные ошибки, адаптирует продукт к новому окружению и вносит небольшие улучшения?

Ответ: Сопровождение

Задание 9

Если вы измеряете процент исходного кода, который выполняется вашими автоматическими тестами, как называется эта метрика качества?

Ответ: Покрытие кода

Задание 10

Какой ключевой антипаттерн в управлении проектами описывает ситуацию, когда новая функция добавляется в проект просто потому, что она кажется интересной, а не потому, что она необходима? («Потому что можем»).

Ответ: Золотая молотка

Задание 11

В модели CI/CD аббревиатура «CI» расшифровывается как...

Ответ: Непрерывная интеграция

Задание 12

Как называется документ или его раздел, который содержит четкие и проверяемые условия, при которых пользовательская история или функция считается завершенной и работоспособной?

Ответ: Критерии приемки

Задание 13

Если база данных хранит данные в виде узлов, связей между ними и их свойств, чтобы эффективно работать со сложными взаимосвязями (например, в соцсетях), то к какому основному типу NoSQL-БД она относится?

Ответ: Графовая

Задание 14

Как называется практика, при которой репозиторий кода имеет одну главную ветку, а вся разработка ведется через создание короткоживущих веток для отдельных задач?

Ответ: Ветвление по функциональности

Задание 15

Какой показатель (KPI) в DevOps измеряет среднее время, которое требуется команде на восстановление работоспособности системы после сбоя?

Ответ: MTTR

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

а) Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

основная

Л1.1 Бабушкина И. А., Окулов С. М. Практикум по объектно-ориентированному программированию [Электронный ресурс]:. - Москва: Лаборатория знаний, 2020. - 369 с. – Режим доступа: <https://e.lanbook.com/book/135561>

Л1.2 Златопольский Д. М. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс]:ВО - Бакалавриат. - Москва: Лаборатория знаний, 2020. - 226 с. – Режим доступа: <https://e.lanbook.com/book/135562>

Л1.3 Бедердинова О. И., Минеева Т. А. Программирование на языках высокого уровня [Электронный ресурс]:учеб. пособие; ВО - Бакалавриат. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2019. - 159 с. – Режим доступа: <http://new.znanium.com/go.php?id=1044396>

Л1.4 Дадян Э. Г. Современные технологии программирования. Язык 1С 8.3 [Электронный ресурс]:учебник; ВО - Бакалавриат. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2023. - 173 с. – Режим доступа: <https://znanium.com/catalog/document?id=426754>

дополнительная

Л2.1 Крищенко В. А., Рязанова Н. Ю. Основы программирования в ядре операционной системы GNU/Linux [Электронный ресурс]:учеб. пособие ; ВО - Бакалавриат. - Москва: МГТУ им. Н.Э. Баумана, 2010. - 34 с. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=58435

Л2.2 Воронцова Е. А. Программирование на С++ с погружением: практические задания и примеры кода [Электронный ресурс]:практикум ; ВО - Бакалавриат. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2016. - 80 с. – Режим доступа: <http://new.znanium.com/go.php?id=563294>

Л2.3 Шишов О. В. Программируемые контроллеры в системах промышленной автоматизации [Электронный ресурс]:учебник ; ВО - Бакалавриат. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2021. - 365 с. – Режим доступа: <http://znanium.com/catalog/document?id=366933>

Л2.4 Жулабова Ф. Т. Системное программирование. Лабораторные работы [Электронный ресурс]:учеб. пособие для СПО. - Санкт-Петербург: Лань, 2021. - 208 с. – Режим доступа: <https://e.lanbook.com/book/164955>

Л2.5 Окулов С. М., Ашихмина Т. В., Бушмелева Н. А. Задачи по программированию [Электронный ресурс]:ВО - Бакалавриат. - Москва: Лаборатория знаний, 2021. - 826 с. – Режим доступа: <https://e.lanbook.com/book/172248>

б) Методические материалы, разработанные преподавателями кафедры по дисциплине, в соответствии с профилем ОП.

Л3.1 Дадян Э. Г. Современные технологии программирования. Язык 1С 8.3 [Электронный ресурс]:учебник; ВО - Бакалавриат. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2025. - 173 с. – Режим доступа: <https://znanium.ru/catalog/document?id=457145>

Л3.2 Богданова С. В., Ермакова А. Н. Технологии программирования:учебник. - Ставрополь, 2024. - 9,60 МБ

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

№	Наименование ресурса сети «Интернет»	Электронный адрес ресурса
1	Введение в программирование	https://intuit.ru/studies/courses/4453/686/info
2	Большой список образовательных ресурсов по программированию	https://intalent.pro/article/bolshoy-spisok-obrazovatelnyh-resursov.html

10. Методические указания для обучающихся по освоению дисциплины

Специфика изучения дисциплины «Технологии программирования» обусловлена формой обучения студентов, ее местом в подготовке бакалавров и временем, отведенным на освоение курса рабочим учебным планом.

Курс обучения делится на время, отведенное для занятий, проводимых в аудиторной форме (лекции, практические и лабораторные занятия) и время, выделенное на внеаудиторное освоение дисциплины, большую часть из которого составляет самостоятельная работа студента.

Лекционная часть учебного курса для студентов проводится в форме обзоров по основным темам. Практические и лабораторные занятия предусмотрены для закрепления теоретических знаний, углубленного рассмотрения наиболее сложных проблем дисциплины, выработки навыков структурно-логического построения учебного материала и отработки навыков самостоятельной подготовки.

Самостоятельная работа студента включает в себя изучение теоретического материала курса, выполнение практических заданий, подготовку к контрольно-обобщающим мероприятиям.

Для освоения курса дисциплины студенты должны:

- изучить материал лекционных и практических занятий в полном объеме по разделам курса;
- выполнить задание, отведенное на самостоятельную работу: подготовить и защитить реферат по утвержденной преподавателем теме, подготовиться к собеседованию, тестированию, защите практических работ, контрольной работе;
- продемонстрировать сформированность компетенций, закрепленных за курсом дисциплины во время мероприятий текущего и промежуточного контроля знаний.

Посещение лекционных и практических занятий для студентов очной и заочной формы является обязательным.

Уважительными причинами пропуска аудиторных занятий является:

- освобождение от занятий по причине болезни, выданное медицинским учреждением,
- распоряжение по деканату, приказ по вузу об освобождении в связи с участием в внутривузовских, межвузовских и пр. мероприятиях,
- официально оформленное свободное посещение занятий.

Пропуски отрабатываются независимо от их причины.

Пропущенные темы лекционных занятий должны быть законспектированы в тетради для лекций, конспект представляется преподавателю для ликвидации пропуска. Пропущенные практические занятия отрабатываются в виде устной защиты практического занятия во время консультаций по дисциплине.

Контроль сформированности компетенций в течение семестра проводится в форме устного опроса на практических занятиях, защиту практических работ, выполнения контрольных работ, написания тестового контроля по теоретическому курсу дисциплины.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства и информационных справочных систем (при необходимости).

11.1 Перечень лицензионного программного обеспечения

1. Kaspersky Total Security - Антивирус
2. Microsoft Windows Server STDCORE AllLngLicense/Software AssurancePack Academic OLV 16Licenses LevelE AdditionalProduct CoreLic 1Year - Серверная операционная система

11.3 Перечень программного обеспечения отечественного производства

1. Kaspersky Total Security - Антивирус

При осуществлении образовательного процесса студентами и преподавателем используются следующие информационно справочные системы: СПС «Консультант плюс», СПС «Гарант».

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

№ п/п	Наименование специальных помещений и помещений для самостоятельной работы	Номер аудитор ии	Оснащенность специальных помещений и помещений для самостоятельной работы
1	Учебная аудитория для проведения занятий всех типов (в т.ч. лекционного, семинарского, практической подготовки обучающихся), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	315/НК 423/НК 423/НК	<p>Оснащение: специализированная мебель на 250 посадочных мест, трибуна для лектора – 1 шт., президиум – 1 шт., видеостена из 9 бесшовный ЖК дисплеев Mercury Full HD 55” ширина-3,1 м высота - 1,7 м , АРМ на основе Intel Core i3 , Монитор Dell 21.5", Клавиатура + мышь , Источник бесперебойного питания 650ВА, Монитор ЖК размер экрана: Dell 21.5", широкоформатная матрица VA с разрешением 1920×1080, отношением сторон 16:9 - 3шт.,микрофонная система Restmoment RX-812 -1шт, Restmoment RX-D58 микрофон делегата -4шт.,АМС настенный громкоговоритель мониторного типа - 6шт., DSPPA микшер-усилитель - 1шт., магнитно-маркерная доска – 1 шт., учебно-наглядные пособия в виде тематических презентаций, информационные плакаты, подключение к сети «Интернет», доступ в электронную информационно-образовательную среду университета, выход в корпоративную сеть университета.</p> <p>Оснащение: специализированная мебель на 56 посадочных мест, стол преподавателя – 1 шт., Sharp 70" Информационный ЖК-дисплей – 1 шт., магнитно-маркерная доска – 1 шт.,</p>
2	Помещение для самостоятельной работы обучающихся, подтверждающее наличие материально-технического обеспечения, с перечнем основного оборудования		
		423/НК	<p>Оснащение: специализированная мебель на 56 посадочных мест, стол преподавателя – 1 шт., Sharp 70" Информационный ЖК-дисплей – 1 шт., магнитно-маркерная доска – 1 шт., учебно-наглядные пособия в виде тематических презентаций, информационные плакаты, подключение к сети «Интернет», доступ в электронную информационно-образовательную среду университета, выход в корпоративную сеть университета.</p>

13. Особенности реализации дисциплины лиц с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники и учебные пособия, иная учебная литература, специальные технические средства обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

а) для слабовидящих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую 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 промежуточной аттестации оформляются увеличенным шрифтом;

- задания для выполнения на промежуточной аттестации 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;

в) для глухих и слабослышащих:

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

- промежуточная аттестация проводится в письменной форме;

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

- по желанию студента промежуточная аттестация может проводиться в письменной форме;

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

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

- по желанию студента промежуточная аттестация проводится в устной форме.

Рабочая программа дисциплины «Технологии программирования» составлена на основе Федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 38.03.05 Бизнес-информатика (приказ Минобрнауки России от 29.07.2020 г. № 838).

Автор (ы)

_____ доцент , к.п.н. Богданова С.В.

Рецензенты

_____ доцент , к.т.н. Шлаев Д.В.

_____ профессор , д.э.н. Шуваев А.В.

Рабочая программа дисциплины «Технологии программирования» рассмотрена на заседании Кафедра информационных систем протокол № 9 от 04.05.2025 г. и признана соответствующей требованиям ФГОС ВО и учебного плана по направлению подготовки 38.03.05 Бизнес-информатика

Заведующий кафедрой _____ Березницкий А.С.

Рабочая программа дисциплины «Технологии программирования» рассмотрена на заседании учебно-методической комиссии Институт экономики, финансов и управления в АПК протокол № от г. и признана соответствующей требованиям ФГОС ВО и учебного плана по направлению подготовки 38.03.05 Бизнес-информатика

Руководитель ОП _____