



**Автономная некоммерческая образовательная организация
высшего образования
«Воронежский экономико-правовой институт»
(АНОО ВО «ВЭПИ»)**

УТВЕРЖДАЮ



**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Б1.О.10 Информатика и программирование
(наименование дисциплины (модуля))

27.03.05 Инноватика
(код и наименование направления подготовки)

Направленность (профиль) Инновационные технологии
(наименование направленности (профиля))

Квалификация выпускника Бакалавр
(наименование квалификации)

Форма обучения Очная, заочная
(очная, заочная)

Рекомендован к использованию Филиалами АНОО ВО «ВЭПИ»

Воронеж 2025

Фонд оценочных средств по дисциплине (модулю) рассмотрен и одобрен на заседании кафедры прикладной информатики.

Протокол от «08» октября 2025 г. № 2

Фонд оценочных средств по дисциплине (модулю) согласован со следующими представителями работодателей или их объединений, направление деятельности которых соответствует области профессиональной деятельности, к которой готовятся обучающиеся:

Директор ООО ПСК «НСКОМ» Петров Р.А.

(должность, название организации, ФИО, печать)



Директор ООО «Сфера АйТи» Павлов С.В.

(должность, название организации, ФИО, печать)



Заведующий кафедрой

М.С. Агафонова

Разработчики:

Ст. преподаватель

Д.В. Байбеков

- для заочной формы обучения:

Наименование дисциплин (модулей), практик	Этапы формирования компетенций по курсам изучения					
	1 курс	2 курс	3 курс	4 курс	5 курс	6 курс
Математика	УК-1	УК-1				
Теория вероятностей и математическая статистика		УК-1				
Системный анализ и принятие решений			УК-1			
Программная инженерия	УК-1					
Информационные системы и технологии		УК-1; ОПК-7; ОПК-10				
Компьютерная и инженерная графика		УК-1				
Стратегический менеджмент в инновационных организациях	УК-1					
Применение Excel в экономических расчетах		УК-1				
Системы искусственного интеллекта в профессиональной деятельности			УК-1; ОПК-7; ОПК-10			
Исследование операций и методы оптимизации			УК-1			
Технические измерения и приборы		УК-1				
Теория алгоритмов		УК-1				
Документирование инновационной деятельности		УК-1				
Делопроизводство		УК-1				
Учебная практика (ознакомительная практика)		УК-1				
Подготовка к сдаче и сдача государственного экзамена						УК-1; ОПК-7; ОПК-10
Выполнение, подготовка к процедуре защиты и защита выпускной квалификационной работы						УК-1; ОПК-7; ОПК-10
Промышленные технологии и инновации			ОПК-7			
Пользовательские аспекты применения средств вычислительной техники	ОПК-7					
Производственная практика (преддипломная практика)						ОПК-10

Этап дисциплины (модуля) Б1.О.09 Информатика и программирование в формировании компетенций соответствует:

- для очной формы обучения – 3,4 семестру;
- для заочной формы обучения – 5 курсу.

2. Показатели и критерии оценивания компетенций на различных этапах их формирования, шкалы оценивания

Показателями оценивания компетенций являются следующие результаты обучения:

Код компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине (модулю)
УК-1. Способен	ИУК 1.1 Анализирует задачу, выделяя ее	Знать: фундаментальные основы высшей

<p>осуществлять поиск, критический анализ и синтез информации, применять системный подход для поставленных задач</p>	<p>базовые составляющие</p>	<p>математики, включая алгебру, геометрию, математический анализ, теорию вероятностей и основы математической статистики, необходимые для анализа задач, возникающих в практической деятельности</p> <p>Уметь: самостоятельно находить математический аппарат, содержащийся в литературе по строительным наукам, и критически анализировать информацию, необходимую для решения поставленной задачи, рассматривать различные варианты решения задачи, оценивая их достоинства и недостатки</p> <p>Владеть: первичными навыками использования математического аппарата для выработки системного подхода к решению поставленных задач</p>
	<p>ИУК 1.2. Находит и критически анализирует информацию, необходимую для решения поставленной задачи</p>	<p>Знать: основные понятия информатики, современные средства вычислительной техники, основы алгоритмического языка и технологию составления программ</p> <p>Уметь: осуществлять поиск, критический анализ и синтез информации на компьютере</p> <p>Владеть: навыками системного подхода для решения поставленных задач</p>
<p>ОПК-7.Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности</p>	<p>ИОПК 7.1 Знает принципы работы современных информационных технологий</p>	<p>Знать: теорию алгоритмов, языки и системы программирования, базы данных; современные тенденции развития информатики и инфокоммуникационных технологий; основные сведения о пакетах прикладных программ, используемых в профессиональной деятельности</p> <p>Уметь: применять средства информатизации, коммуникации и технологии автоматизации для решения прикладных задач</p> <p>Владеть: навыками работы с пакетами прикладных программ для решения профессиональных задач</p>
	<p>ИОПК 7.2 Применяет принципы работы информационных технологий для решения задач профессиональной деятельности</p>	<p>Знать: основы применения ЭВМ для моделирования и оптимизации различных процессов</p> <p>Уметь: применять пакеты прикладных программ для решения прикладных инженернотехнических и технико-экономических задач</p> <p>Владеть: методами применения пакетов прикладных программ для решения задач моделирования и оптимизации</p>
<p>ОПК-10. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</p>	<p>ИОПК 10.1 Применяет алгоритмы и компьютерные программы для решения конкретных задач инновационного развития</p>	<p>Знать: типовые методы численного решения задач оптимизации и принятия решений в условиях неопределенности и риска</p> <p>Уметь: использовать методы теории принятия оптимальных решений для планирования и анализа экономических процессов и систем</p> <p>Владеть: методами и средствами управления информацией в процессе принятия оптимальных решений планирования и анализа экономических процессов и систем</p>
	<p>ИОПК 10.2 Находит способы решения алгоритмов и компьютерных программ в области инновационного проектирования</p>	<p>Знать: алгоритмы для практического применения</p> <p>Уметь: публично представляет результаты решения конкретной задачи программы и проекта</p> <p>Владеть: компьютерными программами для практического применения</p>

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины (модуля):

№ п/п	Наименование раздела дисциплины (модуля)	Код компетенции, код индикатора достижения компетенции	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
1	Тема 1. Основные этапы развития информационного общества	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - предмет изучения Уметь: - работать с информационными ресурсами Владеть: - этапами развития технических средств и информационных ресурсов.	Доклад	«Зачтено» «Не зачтено»
2	Тема 2. Двоичное представление информации	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - подходы к понятию информации и измерению информации. Уметь: - представлять информацию в двоичной системе счисления. Владеть: -	Опрос	«Зачтено» «Не зачтено»
3	Тема 3. Логические основы ЭВМ	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - принципы обработки информации компьютером. Уметь: - решать задачи Владеть: - этапами решения задач с использованием компьютера	Доклад	«Зачтено» «Не зачтено»
4	Тема 4. Основы алгоритмизации и программирования	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - основные элементы языка. Уметь: - описать процедуру и функции. Владеть: - логическими операциями	Сообщение	«Зачтено» «Не зачтено»
5	Тема 5. Линейный алгоритм	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10	Знать: - численные методы решения алгебраических уравнений.	Опрос	«Зачтено» «Не зачтено»

		(ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Уметь: - применять метод Эйлера. Владеть: - метод Рунге-Кутта.		
6	Тема 6. Управляющие конструкции.	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - понятие «операционная система» Уметь: - решать примеры использования внешних устройств, подключаемых к компьютеру, в учебных целях. Владеть: - подключением внешних устройств к компьютеру и их настройкой	Доклад	«Зачтено» «Не зачтено»
7	Тема 7. Циклический алгоритм	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - разграничение прав доступа в сети Уметь: - защитить информацию Владеть: - антивирусной защитой	Опрос	«Зачтено» «Не зачтено»
8	Тема 8. Отладка программы	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - эксплуатационные требования к компьютерному рабочему месту. Уметь: - решать задачи Владеть: - профилактическими мероприятиями для компьютерного рабочего места в соответствии с его комплектацией для профессиональной деятельности.	Опрос	«Зачтено» «Не зачтено»
9	Тема 9. Функции и структура программы	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - программы – переводчики Уметь: - создавать компьютерные публикации на основе использования готовых шаблонов (для выполнения учебных заданий). Владеть: - использование систем проверки	Опрос	«Зачтено» «Не зачтено»

			орфографии и грамматики.		
10	Тема 10. Массивы одномерные и многомерные.	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - систему статистического учета (бухгалтерский учет, планирование и финансы, статистические исследования). Уметь: - использовать различные возможности динамических (электронных) таблиц для выполнения учебных заданий. Владеть: - средствами графического представления статистических данных – деловая графика.	Доклад	«Зачтено» «Не зачтено»
11	Тема 11. Строковые типы данных	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - формирование запросов для работы с электронными каталогами библиотек, музеев, книгоиздания, СМИ в рамках учебных заданий из различных предметных областей. Уметь: - заполнять поля баз данных. Владеть: - формированием запросов для поиска и сортировки информации в базе данных.	Доклад	«Зачтено» «Не зачтено»
12	Тема 12. Указатели и структуры данных.	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - использование презентационного оборудования. Уметь: - создавать и редактировать графические и мультимедийные объекты средствами компьютерных презентаций для выполнения учебных заданий. Владеть: - примерами геоинформационных систем.	Опрос	«Зачтено» «Не зачтено»

13	Тема 13. Работа с файлами.	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - браузер Уметь: - решать примеры работы с Интернет-магазином, Интернет-СМИ, Интернет-турагентством, Интернет-библиотекой Владеть: - методами и средствами создания и сопровождения сайта.	Опрос	«Зачтено» «Не зачтено»
14	Тема 14. Сортировка данных.	УК-01 (ИУК-1.1, ИУК-1.2.) ОПК-10 (ИОПК-10.1, ИОПК-10.2) ОПК-7 (ИОПК-7.1, ИОПК-7.2)	Знать: - сетевые информационные системы для различных направлений профессиональной деятельности Уметь: - использовать тестирующие системы в учебной деятельности в локальной сети образовательного учреждения. Владеть: - участием в онлайн конференции, анкетировании, конкурсе, олимпиаде или тестировании.	Доклад	«Зачтено» «Не зачтено»
ИТОГО			Форма контроля	Оценочные средства промежуточной аттестации	Шкала оценивания
			Экзамен	Письменный ответ на билет	«Отлично», «Хорошо», «Удовлетворительно», «Неудовлетворительно»

Критерии оценивания результатов обучения для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

1. Критерий оценивания опроса:

- зачтено – выставляется обучающемуся, если демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки; освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при ответе; достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности; показывает

всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их;

- не зачтено – выставляется обучающемуся, если демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки; допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки; выставляется обучающемуся, ответ которого содержит существенные пробелы в знаниях основного содержания рабочей программы дисциплины.

2. Критерий доклада:

- зачтено – представленный доклад соответствует тематике, экономически обоснован, выводы по изученной проблеме изложены логически, соблюдены требования, при разработке доклада были использованы современные информационные технологии;

- не зачтено – доклад обучающимся не представлена; материалы доклад не обоснованы или логически не связаны, использованы устаревшие источники информации.

3. Критерий сообщения:

- зачтено – представленный сообщение актуально, экономически обоснован, выводы по изученной представленная информация изложена логически, соблюдены требования, при разработке сообщения были использованы современные информационные технологии;

- не зачтено – сообщение обучающимся не представлена; представленная информация не обоснованы или логически не связана, использованы устаревшая информация.

4. Критерии оценивания письменного ответа на билет на экзамене:

- отлично – выставляется обучающемуся, если: даны исчерпывающие и обоснованные ответы на все поставленные вопросы, правильно и рационально (с использованием рациональных методик) решены соответствующие задачи; в ответах выделялось главное, все теоретические положения умело увязывались с требованиями руководящих документов; ответы были четкими и краткими, а мысли излагались в логической последовательности; показано умение самостоятельно анализировать факты, события, явления, процессы в их взаимосвязи и диалектическом развитии;

- хорошо – выставляется обучающемуся, если: даны полные, достаточно обоснованные ответы на поставленные вопросы, правильно решены практические задания; в ответах не всегда выделялось главное, отдельные положения недостаточно увязывались с требованиями руководящих документов, при решении практических задач не всегда

использовались рациональные методики расчётов; ответы в основном были краткими, но не всегда четкими; показано слабое умение самостоятельно анализировать факты, события, явления, процессы в их взаимосвязи и диалектическом развитии;

- удовлетворительно – выставляется обучающемуся, если: даны в основном правильные ответы на все поставленные вопросы, но без должной глубины и обоснования, при решении практических задач обучающийся использовал прежний опыт и не применял новые методики выполнения расчётов, однако на уточняющие вопросы даны в целом правильные ответы; при ответах не выделялось главное; отдельные положения недостаточно увязывались с требованиями руководящих документов, при решении практических задач не использовались рациональные методики расчётов; ответы были многословными, нечеткими и без должной логической последовательности, на отдельные дополнительные вопросы не даны положительные ответы; показано неумение самостоятельно анализировать факты, события, явления, процессы в их взаимосвязи и диалектическом развитии.

- неудовлетворительно – выставляется обучающемуся, если не выполнены требования, соответствующие оценке “удовлетворительно”.

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

3.1. Вопросы для проведения опроса:

1. Предмет изучения, последовательность сдачи экзамена; рекомендации по самостоятельной работе над учебным материалом.
2. Роль информационной деятельности в современном обществе: экономической, социальной, культурной, образовательной сферах.
3. Этапы развития технических средств и информационных ресурсов.
4. Информационные ресурсы общества.
5. Подходы к понятию информации и измерению информации.
6. Информационные объекты различных видов.
7. Универсальность дискретного (цифрового) представления информации.
8. Принципы обработки информации компьютером.
9. Арифметические и логические основы работы компьютера.
10. Алгоритмы и способы их описания.
11. Этапы решения задач с использованием компьютера: формализация, программирование и тестирование.
12. Переход от неформального описания к формальному.
13. Оператор цикла с предпроверкой условия.
14. Оператор цикла с постпроверкой условия.
15. Вложенные циклы.

16. Описание процедур и функций.
17. Параметры. Обращение к подпрограммам.
18. Обмен информацией между программными единицами.
19. Перечисляемый цикл.
20. Строковые типы данных и их обработка.
21. Комментарии однострочные и многострочные.
22. Условный алгоритм.
23. Программное обеспечение внешних устройств.
24. Подключение внешних устройств к компьютеру и их настройка.
25. Разграничение прав доступа в сети, общее дисковое пространство в локальной сети.
26. Защита информации, антивирусная защита.
27. Эксплуатационные требования к компьютерному рабочему месту.
28. Профилактические мероприятия для компьютерного рабочего места в соответствии с его комплектацией для профессиональной деятельности.
29. Использование систем проверки орфографии и грамматики.
30. Создание компьютерных публикаций на основе использования готовых шаблонов (для выполнения учебных заданий).
31. Программы – переводчики.
32. Возможности систем распознавания текстов.
33. Гипертекстовое представление информации.
34. Правила объявления и инициализации переменных.
35. Массивы одномерные и многомерные.
36. Указатели на область памяти переменных.
37. Интерпретатор, компилятор, транслятор.
38. Синтаксические ошибки в программе.
39. Семантические ошибки в программе.
40. Основные типы переменных в си++.
41. Отладка программы.
42. Что такое тело программы.
43. Сложные логические условия в алгоритме выбора.
44. Функции, возвращающие значения. Формат и особенности.
45. Функции, НЕ возвращающие значения. Формат и особенности.
46. Классификация языков программирования.
47. Языки высокого уровня. Примеры, особенности, назначение.
48. Языки низкого уровня. Примеры, особенности, назначение.
49. Виды представления алгоритмов.
50. Псевдокод в программировании. Назначение, особенности.
51. Классические библиотеки си++. Способы подключения.

3.2. Примерный перечень тем докладов и сообщений:

1. Роль информационной деятельности в современном обществе: экономической, социальной, культурной, образовательной сферах.
2. Этапы развития технических средств и информационных ресурсов.

3. Виды профессиональной информационной деятельности человека с использованием технических средств и информационных ресурсов социально-экономической деятельности (специальное ПО, порталы, БД).

4. Универсальность дискретного (цифрового) представления информации.

5. Дискретное (цифровое) представление текстовой, графической, звуковой информации и видеоинформации.

6. Арифметические и логические основы работы компьютера.

7. Алгоритмы и способы их описания.

8. Этапы решения задач с использованием компьютера: формализация, программирование и тестирование.

9. Обмен информацией между программными единицами.

10. Графический интерфейс пользователя.

11. Примеры использования внешних устройств, подключаемых к компьютеру, в учебных целях.

12. Примеры комплектации компьютерного рабочего места в соответствии с целями его использования для различных направлений профессиональной деятельности.

13. Защита информации, антивирусная защита.

14. Профилактические мероприятия для компьютерного рабочего места в соответствии с его комплектацией для профессиональной деятельности.

15. Использование различных возможностей динамических (электронных) таблиц для выполнения учебных заданий.

16. Системы статистического учета (бухгалтерский учет, планирование и финансы, статистические исследования).

17. Представление результатов выполнения расчетных задач средствами деловой графики.

18. Формирование запросов для работы с электронными каталогами библиотек, музеев, книгоиздания, СМИ в рамках учебных заданий из различных предметных областей.

19. Формирование запросов для поиска и сортировки информации в базе данных.

20. Создание и редактирование графических и мультимедийных объектов средствами компьютерных презентаций для выполнения учебных заданий.

21. Примеры работы с Интернет-магазином, Интернет-СМИ, Интернет-турагентством, Интернет-библиотекой и пр.

22. Методы и средства создания и сопровождения сайта.

23. Использование тестирующих систем в учебной деятельности в локальной сети образовательного учреждения.

24. Возможности систем управления базами данных.

25. Базы данных.

3.3. Вопросы для проведения экзамена:

1. Интегрированная среда си/си++: принципы взаимодействия с системой. Вызов системы, главное меню, подменю (краткое описание).
2. Алфавит языка си/си++. Лексические элементы языка си/си+..
Метаязыки для описания синтаксиса языков программирования: расширенные БНФ, синтаксические диаграммы.
3. Концепция типов данных. Иерархия типов данных в языке си/си++.
Примеры использования стандартных типов данных.
4. Программирование выражений в си/си++. Оператор присваивания.
5. Условный оператор (if -else) и оператор выбора (switch).
6. Оператор цикла: с предусловием (whileo), с постусловием (do-while), с параметром (for).
7. Множественный тип данных: определение и примеры использования.
8. Регулярный и строковый типы данных: определение и примеры использования.
9. Комбинированный тип данных. Фиксированные записи и записи с вариантной частью. Оператор присоединения (with).
10. Текстовые файлы в си/си++. Процедуры и функции для работы с текстовыми файлами. Пример программы, использующей текстовые файлы.
11. Рекурсия. Примеры рекурсивных процедур и функций.
12. Понятие подпрограммы. Подпрограммы в языке си/си++: процедуры и функции. Блочная структура программ на языке си/си++.
13. Механизмы передачи параметров. Передача параметров по значению. Передача параметров по ссылке.
14. Сущность объектно-ориентированного подхода в программировании на си/си++. Инкапсуляция, наследование, полиморфизм.
15. Введение в алгоритмы сортировки массивов.

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Экзамен является заключительным этапом процесса формирования компетенций обучающегося при изучении дисциплины и имеет целью проверку и оценку знаний обучающегося по теории и применению полученных знаний, умений и навыков при решении практических задач.

Экзамен проводится по расписанию, сформированному учебно-методическим управлением, в сроки, предусмотренные календарным учебным графиком.

Экзамен принимается преподавателем, ведущим лекционные занятия.

Экзамен проводится только при предъявлении обучающимся зачетной книжки и при условии выполнения всех контрольных мероприятий, предусмотренных учебным планом и рабочей программой дисциплины.

Обучающимся на экзамене представляется право выбрать один из билетов. Время подготовки к ответу составляет 30 минут. По истечении установленного времени обучающийся должен ответить на вопросы экзаменационного билета.

Результаты экзамена оцениваются по четырехбалльной системе и заносятся в зачетно-экзаменационную ведомость и зачетную книжку. В зачетную книжку заносятся только положительные оценки. Подписанный преподавателем экземпляр ведомости сдаётся не позднее следующего дня в деканат.

В случае неявки обучающегося на экзамен в зачетно-экзаменационную ведомость делается отметка «не явка».

Обучающиеся, не прошедшие промежуточную аттестацию по дисциплине, должны ликвидировать академическую задолженность в установленном локальными нормативными актами Института порядке.

5. Материалы для компьютерного тестирования обучающихся в рамках проведения контроля наличия у обучающихся сформированных результатов обучения по дисциплине

Общие критерии оценивания

№ п/п	Процент правильных ответов	Оценка
1	86 % – 100 %	5 («отлично»)
2	70 % – 85 %	4 («хорошо»)
3	51 % – 69 %	3 («удовлетворительно»)
4	50 % и менее	2 («неудовлетворительно»)

Вариант 1

Номер вопроса и проверка сформированной компетенции

№ вопроса	Код компетенции	№ вопроса	Код компетенции
1	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	11	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
2	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	12	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
3	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	13	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
4	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	14	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
5	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	15	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
6	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	16	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
7	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	17	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
8	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	18	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
9	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	19	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
10	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	20	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)

Ключ ответов

№ вопроса	Верный ответ	№ вопроса	Верный ответ
1	2	11	4
2	4	12	3
3	3	13	3
4	3	14	1
5	4	15	4
6	1	16	3
7	2	17	2
8	3	18	1
9	4	19	2
10	3	20	3

Задание № 1.

Теоретическая информатика опирается на

Ответ:

1. Законы природы;
- 2.** Математическую логику, теорию алгоритмов, теорию кодирования, системный анализ;
3. Разделы математики: численный анализ, математический анализ, дифференциальные уравнения.
4. На законы механики и электричества;

Задание № 2.

В состав программного обеспечения ЭВМ не входят

Ответ:

1. Прикладные программы.
2. Системы программирования;
3. Операционные системы;
- 4.** Аппаратные средства;

Задание № 3.

Операционная система DOS является

Ответ:

1. Многопользовательской, многозадачной.
2. Многопользовательской, однозадачной;
- 3.** Однопользовательской, однозадачной;
4. Однопользовательской, многозадачной;

Задание № 4.

Все существующие языки программирования делятся на

Ответ:

1. Функциональные и логические
2. Русско- и нерусскоязычные
- 3.** Языки низкого и высокого уровня.

Задание № 5.

Абзацные отступы и ширина колонок могут изменяться в Word с помощью

Ответ:

1. Линейки прокрутки;
2. Заголовка окна
3. Строки состояния;
- 4.** Координатной линейки;

Задание № 6.

Текстовый редактор Word – это

Ответ:

- 1.** Прикладная программа;
2. Базовое программное обеспечение;
3. Сервисная программа;
4. Редактор шрифтов;

Задание № 7.

При нажатии на кнопку с изображением дискеты на панели пиктографического меню в Word происходит

Ответ:

1. Запись документа на дискету;
- 2.** Сохранение документа;
3. Считывание информации с дискеты;
4. Печать документа.

Задание № 8.

Издательская система представляет собой

Ответ:

1. Систему управления базой данных;
2. Операционную оболочку;
- 3.** Комплекс аппаратных и программных средств;
4. Графический редактор

Задание № 9.

Способ реализации построения изображений на экране дисплея, при котором электронный луч поочередно рисует на экране различные знаки - элементы изображения, называется

Ответ:

1. Растровым;
2. Графическим.
3. Лучевым;
- 4. Векторным;**

Задание № 10.

Структура данных, для которой характерна подчиненность объектов нижнего уровня объектам верхнего уровня, называется

Ответ:

1. Табличной;
2. Реляционной;
- 3. Иерархической;**
4. Сетевой.

Задание № 11.

Отличительной чертой реляционной базы данных является

Ответ:

1. Подчиненность объектов нижнего уровня объектам верхнего уровня;
2. Возможность поиска данных по ключу;
3. То, что отношения между объектами определяются как "многие ко многим";
- 4. То, что каждая запись в таблице содержит информацию, относящуюся только к одному конкретному объекту.**

Задание № 12.

Антивирусные средства предназначены

Ответ:

1. Для тестирования системы;
2. Для защиты программ от вируса;
- 3. Для проверки программ на наличие вируса и их лечения;**
4. Для мониторинга системы.

Задание № 13.

Оператор - это

Ответ:

1. Законченная фраза языка, предписание, команда;
2. Алгоритм действия программы, написанной на данном языке;
3. Процедура обработки данных.
4. Функция, которая оперирует с данными;

Задание № 14.

Сигнал – это

Ответ:

1. Сообщение, передаваемое с помощью носителя;
2. Виртуальный процесс передачи информации;
3. Электромагнитный импульс;
4. Световая вспышка.

Задание № 15.

Примером дискретного сигнала является

Ответ:

1. Музыка;
2. Человеческая речь;
3. Видеоинформация;
4. Текстовая информация.

Задание № 16.

Компьютерная сеть-это

Ответ:

1. Мультимедийный компьютер с принтером, модемом и факсом.
2. Группа компьютеров, размещенных в одном помещении;
3. Объединение нескольких ЭВМ для совместного решения задач;
4. Комплекс терминалов, подключенных каналами связи к большой машине;

Задание № 17.

Сетевые технологии – это

Ответ:

1. Способ соединения компьютеров в сети.
2. Технологии обработки информации в компьютерных сетях;
3. Основная характеристика компьютерных сетей;
4. Формы хранения информации;

Задание № 18.

К системному программному обеспечению относятся

Ответ:

- 1.** Новые языки программирования и компиляторы к ним, интерфейсные системы;
2. Системы обработки текстов, электронные процессоры, базы данных;
3. Решение вопросов об анализе потоков информации в различных сложных системах;
4. Поисковые системы, глобальные системы хранения и поиска информации.

Задание № 19.

Назначение программ оболочек

Ответ:

1. Предоставление возможности написания программ;
- 2.** Облегчение взаимодействия пользователя с компьютером;
3. Защита операционной системы;

Задание № 20.

Поименованная совокупность данных, хранимая во внешней памяти, - это

Ответ:

1. Файловая система;
2. Директорий;
- 3.** Файл;
4. Запись

Задания открытого типа (типовые задания, ситуационные задачи)

Номер вопроса и проверка сформированной компетенции

№ вопроса	Код компетенции	№ вопроса	Код компетенции
1	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	11	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
2	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	12	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
3	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	13	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
4	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	14	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)

5	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	15	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
6	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	16	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
7	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	17	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
8	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	18	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
9	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	19	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)
10	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)	20	ОПК-7 (ИОПК-7.1, ИОПК-7.2) УК-1 (ИУК-1.1, ИУК-1.2)

Ключ ответов к заданиям открытого типа

№ вопроса	Верный ответ
1	classify возвращает метку класса (str/int), predict — вещественное число. В k-NN для классификации используется мода соседей, для регрессии — среднее/медиана. Метрики: Евклидова, Манхэттенская, Чебышева.
2	python def preprocess_data(df): from sklearn.impute import SimpleImputer from sklearn.preprocessing import StandardScaler, OneHotEncoder # ... импутация, кодирование, масштабирование return X_processed Используются библиотеки: pandas, sklearn.
3	python def knn_bruteforce(X_train, y_train, X_test, k, metric='euclidean'): predictions = [] for x in X_test: distances = [distance(x, x_train) for x_train in X_train] idx = np.argsort(distances)[:k] pred = stats.mode(y_train[idx])[0][0] predictions.append(pred) return predictions Сложность: $O(n * m * d)$, где n — размер теста, m — размер обучения, d — размерность.
4	Блок-схема включает этапы: Load Data → Preprocess → Split (train/test) → Choose Algorithm (if classification → SVM/kNN, if regression → Linear Reg, if clustering → K-Means) → Train → Evaluate → Deploy.
5	python from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix def binary_classifier_evaluation(y_true, y_pred): return { 'accuracy': accuracy_score(y_true, y_pred), 'precision': precision_score(y_true, y_pred), 'recall': recall_score(y_true, y_pred), 'f1': f1_score(y_true, y_pred), 'confusion_matrix': confusion_matrix(y_true, y_pred) }
6	python def linear_regression_gradient_descent(X, y, lr=0.01, epochs=1000): m, n = X.shape w = np.zeros(n) loss_history = [] for epoch in range(epochs): y_pred = X.dot(w) error = y_pred - y grad = (2/m) * X.T.dot(error) w -= lr * grad loss = np.mean(error**2) loss_history.append(loss) return w, loss_history
7	python # Псевдокод построения KD-Tree def build_kdtree(points, depth=0): if not points: return None axis = depth % k points.sort(key=lambda x: x[axis]) median = len(points)//2 node = Node(points[median]) node.left = build_kdtree(points[:median], depth+1) node.right = build_kdtree(points[median+1:], depth+1) return node Сложность поиска: $O(\log n)$ в среднем, $O(n)$ в худшем.
8	При увеличении n в 10 раз и d в 10 раз, время выполнения увеличится примерно в 100 раз. $O(n * m * d) \rightarrow c 1000 * 1000 * 10$ до $10000 * 10000 * 100 =$ в 10000 раз больше операций.
9	Скрипт использует MinMaxScaler и StandardScaler из sklearn. Результаты: на нормализованных данных ассурасу k-NN обычно выше, так как все признаки вносят равный вклад в расстояние.
10	python def normal_equation(X, y): return np.linalg.pinv(X.T @ X) @ X.T @ y Аналитическое решение быстрее для небольших матриц, но $O(n^3)$ по вычислениям и требует много памяти при больших n . Градиентный спуск масштабируется лучше.
11	python def mse(y_true, y_pred): return np.mean((y_true - y_pred)**2) def mae(y_true, y_pred): return np.mean(np.abs(y_true - y_pred)) def plot_errors(y_true, y_pred):

№ вопроса	Верный ответ
	<code>errors = y_true - y_pred
plt.hist(errors, bins=30); plt.show()
</code>
12	<code>python
def r2_score(y_true, y_pred):
 ss_res = np.sum((y_true - y_pred)**2)
 ss_tot = np.sum((y_true - np.mean(y_true))**2)
 return 1 - ss_res/ss_tot
</code> Если R^2 на обучении близок к 1, а на тесте значительно ниже (например, 0.2) — модель переобучена.
13	<code>python
def kmeans_iteration(points, centroids):
 labels = []
 for p in points:
 distances = [np.linalg.norm(p - c) for c in centroids]
 labels.append(np.argmin(distances))
 new_centroids = []
 for i in range(len(centroids)):
 cluster_points = [points[j] for j in range(len(points)) if labels[j] == i]
 new_centroids.append(np.mean(cluster_points, axis=0))
 return np.array(labels), np.array(new_centroids)
</code>
14	<code>python
def dbscan(X, eps, min_samples):
 # Реализация через Union-Find или поиск в глубину
 labels = [-1]*len(X) # -1 для шума
 cluster_id = 0
 for i, point in enumerate(X):
 if labels[i] != -1: continue
 neighbors = find_neighbors(X, i, eps)
 if len(neighbors) < min_samples:
 labels[i] = -1 # шум
 else:
 expand_cluster(X, labels, i, neighbors, cluster_id, min_samples)
 cluster_id += 1
 return labels
</code>
15	<code>python
from scipy.cluster.hierarchy import linkage, dendrogram
linkage_matrix = linkage(X, method='complete')
dendrogram(linkage_matrix)
plt.show()
</code>
16	<code>python
def build_decision_tree(data, labels, depth=0, max_depth=5):
 if depth == max_depth or len(set(labels)) == 1:
 return LeafNode(majority_class(labels))
 feature, threshold = find_best_split(data, labels) # по критерию Джини
 left_mask = data[:, feature] < threshold
 right_mask = ~left_mask
 left_subtree = build_decision_tree(data[left_mask], labels[left_mask], depth+1)
 right_subtree = build_decision_tree(data[right_mask], labels[right_mask], depth+1)
 return DecisionNode(feature, threshold, left_subtree, right_subtree)
</code>
17	<code>python
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
gb = GradientBoostingClassifier(n_estimators=100).fit(X_train, y_train)
dt = DecisionTreeClassifier(max_depth=5).fit(X_train, y_train)
GradientBoosting обычно даёт accuracy выше на 3-5%, но обучается дольше.
</code>
18	<code>python
def evaluate_model(model, X_test, y_test, task='classification'):
 y_pred = model.predict(X_test)
 if task == 'classification':
 print(classification_report(y_test, y_pred))
 fpr, tpr, _ = roc_curve(y_test, y_pred)
 plt.plot(fpr, tpr); plt.show()
 else:
 print(f'MSE: {mse(y_test, y_pred):.3f}')
 plt.scatter(y_pred, y_test - y_pred); plt.show()
</code>
19	<code>python
import cvxopt
def svm_linear(X, y):
 n = X.shape[0]
 P = cvxopt.matrix(np.outer(y, y) * (X @ X.T))
 q = cvxopt.matrix(-np.ones(n))
 G = cvxopt.matrix(-np.eye(n))
 h = cvxopt.matrix(np.zeros(n))
 A = cvxopt.matrix(y, (1, n), 'd')
 b = cvxopt.matrix(0.0)
 sol = cvxopt.solvers.qp(P, q, G, h, A, b)
 alphas = np.array(sol['x'])
 # Вычисление w и b
 w = np.sum(alphas * y[:, None] * X, axis=0)
 return w, b
</code>
20	<code>python
from sklearn.svm import SVC
model = SVC(kernel='rbf', gamma='scale')
model.fit(X_train, y_train)
</code> Или собственная реализация SMO.

1. Напишите псевдокод двух функций: `classify(x, data, k)` для задачи классификации и `predict(x, data, k)` для задачи регрессии методом k-NN. Объясните различия в возвращаемых значениях и метриках расстояния.

2. Разработайте функцию `preprocess_data(df)` на Python, которая принимает DataFrame и выполняет: обработку пропусков (заполнение медианой для числовых и модой для категориальных признаков), One-Hot Encoding категориальных переменных и стандартизацию числовых. Укажите библиотеки, которые вы бы использовали.

3. Реализуйте алгоритм k-NN «в лоб» (brute-force) на Python. Функция должна принимать матрицу признаков `X_train`, вектор меток `y_train`, параметр `k` и метрику расстояния (евклидову или манхэттенскую). Оцените вычислительную сложность вашей реализации.

4. Создайте блок-схему (flowchart), иллюстрирующую процесс решения задачи машинного обучения: от загрузки данных до оценки модели, с ветвлением на задачи классификации, регрессии и кластеризации.
5. Напишите функцию на Python `binary_classifier_evaluation(y_true, y_pred)`, которая возвращает словарь с Accuracy, Precision, Recall, F1-score и матрицей ошибок. Протестируйте её на простом примере.
6. Реализуйте функцию `linear_regression_gradient_descent(X, y, learning_rate, epochs)`, которая обучает модель линейной регрессии с использованием градиентного спуска. Функция должна возвращать вектор весов w и историю значений функции потерь (MSE).
7. Оптимизируйте вашу реализацию k-NN из задания 3, используя структуру данных KD-Tree для ускорения поиска ближайших соседей. Напишите псевдокод построения KD-Tree и поиска k ближайших соседей.
8. Проанализируйте, как изменится время выполнения вашей функции k-NN при увеличении размера обучающей выборки с 1000 до 10 000 объектов и размерности признаков с 10 до 100. Дайте оценку в нотации O-большое.
9. Напишите скрипт на Python, который загружает датасет с признаками разного масштаба (например, рост в см и доход в рублях), применяет min-max нормализацию и стандартизацию (Z-score), а затем сравнивает результаты классификации k-NN на исходных и преобразованных данных.
10. Реализуйте метод `predict` для модели линейной регрессии, обученной с помощью нормального уравнения (аналитическое решение). Сравните производительность с градиентным спуском на синтетических данных.
11. Напишите функцию `mse(y_true, y_pred)` и `mae(y_true, y_pred)` для расчёта метрик регрессии. Добавьте в функцию возможность визуализировать распределение ошибок с помощью matplotlib.
12. Реализуйте вычисление коэффициента детерминации R^2 . Объясните, какие значения этой метрики на тестовой выборке будут свидетельствовать о переобучении модели.
13. Реализуйте одну итерацию алгоритма K-Means на Python: функция принимает точки данных и текущие центроиды, возвращает новые метки кластеров для каждой точки и новые центроиды.

14. Напишите функцию `dbscan(X, eps, min_samples)`, реализующую алгоритм DBSCAN. Используйте структуру данных графа или Union-Find для объединения точек в кластеры. Визуализируйте результат на 2D данных.

15. Реализуйте агломеративную кластеризацию «снизу вверх». Используйте матрицу расстояний между точками и метод полной связи (complete linkage). Функция должна возвращать дендрограмму.

16. Реализуйте рекурсивную функцию `build_decision_tree(data, labels, depth)` для построения бинарного решающего дерева, использующую критерий Джини для выбора разделения. Ограничьте глубину дерева.

17. Используя библиотеку `scikit-learn`, обучите модель градиентного бустинга (`GradientBoostingClassifier`) на датасете `breast_cancer`. Сравните её accuracy и время обучения с одним решающим деревом (`DecisionTreeClassifier`).

18. Напишите функцию `evaluate_model(model, X_test, y_test)`, которая для задачи классификации выводит `classification report` и строит ROC-кривую. Для задачи регрессии выводит MSE, MAE, R^2 и график `residuals`.

19. Реализуйте SVM с линейным ядром «в лоб» для бинарной классификации, используя библиотеку `svxort` для решения задачи квадратичного программирования (двойственная форма).

20. Напишите класс `KernelSVM`, который принимает на вход ядерную функцию (RBF, полиномиальную) и обучается методом SMO (Sequential Minimal Optimization) или с помощью библиотеки `sklearn`.

Вариант 2**Номер вопроса и проверка сформированной компетенции**

№ вопроса	Код компетенции	№ вопроса	Код компетенции
1	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	11	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
2	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	12	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
3	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	13	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
4	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	14	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
5	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	15	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
6	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	16	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
7	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	17	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
8	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	18	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
9	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	19	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
10	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	20	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)

Ключ ответов

№ вопроса	Верный ответ	№ вопроса	Верный ответ
1	2	11	4
2	4	12	4
3	1	13	4
4	1	14	3
5	2	15	2
6	4	16	1
7	2	17	2
8	1	18	2
9	2	19	4
10	1	20	4

Задание № 1.

Чтобы изменить структуру или шаблон формы в СУБД Access, нужно открыть форму в режиме

Ответ:

1. Таблицы;
- 2.** Конструктора;
3. Формы;
4. Предварительного просмотра

Задание № 2.

Excel – это

Ответ:

1. СУБД;
2. Текстовые редакторы;
3. Графические редакторы;
- 4.** Электронные таблицы.

Задание № 3.

Какое из следующих качеств необязательно присуще программе-вирусу

Ответ:

- 1.** Самостоятельно запускается;
2. Присоединяет свой код к кодам других программ;
3. Занимает малый объем памяти;
4. Приводит к потере информации.

Задание № 4.

Переменная – это

Ответ:

- 1.** Объект, способный принимать различные значения;
2. Динамический объект;
3. Значения чисел;
4. Меняющееся число;

Задание № 5.

Сигнал будет дискретным в случае

Ответ:

1. Когда источник вырабатывает непрерывное сообщение;
- 2.** Когда параметр сигнала принимает последовательное во времени конечное число значений;
3. Когда передается с помощью волны;

4. Когда источником посылается всего один бит/с.

Задание № 6.

Информационные системы – это

Ответ:

1. Компьютерные сети;
2. Хранилище информации;
3. Системы, управляющие работой компьютера;
- 4.** Системы хранения, обработки и передачи информации в специально организованной форме.

Задание № 7.

Сервер – это

Ответ:

1. Мультимедийный компьютер с модемом.
- 2.** Один или несколько мощных компьютеров для обслуживания сети;
3. Высокопроизводительный компьютер;
4. Хранитель программы начальной загрузки;

Задание № 8.

К прикладному программному обеспечению относятся

Ответ:

- 1.** Системы обработки текстов, электронные процессоры, базы данных;
2. Решение вопросов об анализе потоков информации в различных сложных системах;
3. Поисковые системы, глобальные системы хранения и поиска информации.
4. Новые языки программирования и компиляторы к ним, интерфейсные системы;

Задание № 9.

Функция в языке программирования – это

Ответ:

1. Программный объект, принимающий значение с помощью оператора присваивания;
- 2.** Программный объект, задающий вычислительную процедуру определения значения от аргумента;
3. Сегмент программы, хранящий некоторое значение, зависящее от аргумента;
4. Выражение, означающее зависимость левой части от правой.

Задание № 10.

Основными компонентами в составе ОС являются

Ответ:

1. Утилиты, командный процессор, ядро
2. Резидентные программы, ядро, командный процессор, материнская плата
3. Резидентные программы, программы оболочки, утилиты
4. Утилиты, командный процессор, центральный процессор

Задание № 11.

Компоновкой называется

Ответ:

1. Процесс описания переменных в программе;
2. Проверка, не нарушены ли формальные правила, содержащиеся в данном языке программирования;
3. Просмотр и изменение значений переменных в ходе отладки програMif
4. Подключение к исходному объектному модулю объектных модулей соответствующих подпрограмм.

Задание № 12.

Следующая последовательность действий:

выделить нужный участок текста; нажать на нем левую клавишу мыши и, удерживая ее, передвигать мышь до нужного места Word приведет

Ответ:

1. К копированию выделенного участка текста;
2. К удалению выделенного участка текста в буфер.
3. К замене текущего текста на выделенный;
4. К перемещению выделенного текста в новое место;

Задание № 13.

В текстовых редакторах и настольных издательских системах, как правило, с помощью клавиш Alt + F4 происходит

Ответ:

1. Переход в окно с предыдущей программой;
2. Переход в окно со следующей программой;
3. Открытие файла;
4. Выход из программы.

Задание № 14.

Следующая последовательность действий:
установить указатель мыши на полосе выделения рядом с текстом;
нажать левую клавишу мыши и, удерживая ее, передвигать мышь в нужном направлении приведет

Ответ:

1. К удалению текста;
2. К перемещению текста;
- 3.** К выделению текста;
4. К копированию текста в буфер.

Задание № 15.

Способ реализации построения изображений на экране дисплея, при котором изображение представлено прямоугольной матрицей точек, имеющих свой цвет из заданной палитры, называется

Ответ:

- 1. Растровым;**
2. Мозаичным;
3. Пиксельным;
4. Графическим.

Задание № 16.

Макрос – это

Ответ:

- 1.** Объект, представляющий собой структурированное описание одного нескольких действий;
2. Текстовый редактор.
3. Язык программирования;
4. Часть командного процессора;

Задание № 17.

К основным функциям СУБД не относится

Ответ:

1. Определение данных;
- 2.** Хранение данных;
3. Обработка данных;
4. Управление данными

Задание № 18 .

В ячейку электронной таблицы нельзя ввести

Ответ:

1. Формулу;
2. Иллюстрацию;
3. Текст;
4. Число.

Задание № 19.

Какая из ниже перечисленных программ не является антивирусным средством

Ответ:

1. Aidstest;
2. Doctor Web;
3. VSAFE;
4. Vsearch.

Задание № 20.

Модуль – это

Ответ:

1. Набор символов и идентификаторов;
2. Специальная программная единица для создания библиотек;
3. Вспомогательная процедура.
4. Отдельная программа, которая взаимодействует с другими программами;

Задания открытого типа (типовые задания, ситуационные задачи)

Номер вопроса и проверка сформированной компетенции

№ вопроса	Код компетенции	№ вопроса	Код компетенции
1	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	11	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
2	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	12	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
3	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	13	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
4	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	14	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
5	ОПК-7 (ИОПК-7.1, ИОПК-7.2)	15	ОПК-7 (ИОПК-7.1, ИОПК-7.2)

	ОПК-10 (ИОПК-10.1, ИОПК-10.2)		ОПК-10 (ИОПК-10.1, ИОПК-10.2)
6	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	16	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
7	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	17	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
8	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	18	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
9	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	19	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)
10	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)	20	ОПК-7 (ИОПК-7.1, ИОПК-7.2) ОПК-10 (ИОПК-10.1, ИОПК-10.2)

Ключ ответов к заданиям открытого типа

№ вопроса	Верный ответ
1	<pre>java
public class MLAlgorithms {
 public static int knnClassify(double[][] train, int[] labels, double[] sample, int k) {
 // Возвращает int - метку класса
 }
 public static double knnRegress(double[][] train, double[] targets, double[] sample, int k) {
 // Возвращает double - прогнозируемое значение
 }
}</pre>
2	<pre>``csharp using MathNet.Numerics.Statistics; public class DataPreprocessor { public void RemoveOutliers(List<double> data) { double q1 = data.Quantile(0.25); double q3 = data.Quantile(0.75); double iqr = q3 - q1; data.RemoveAll(x => x < q1 - 1.5*iqr</pre>
3	<pre>c++
#include <queue>
#include <vector>
#include <cmath>
double euclidean_dist(const vector<double>& a, const vector<double>& b) {
 double sum = 0.0;
 for(size_t i=0; i<a.size(); ++i) sum += pow(a[i]-b[i], 2);
 return sqrt(sum);
}
double cosine_sim(const vector<double>& a, const vector<double>& b) {
 double dot = 0.0, norm_a = 0.0, norm_b = 0.0;
 for(size_t i=0; i<a.size(); ++i) {
 dot += a[i]*b[i]; norm_a += a[i]*a[i]; norm_b += b[i]*b[i];
 }
 return dot/(sqrt(norm_a)*sqrt(norm_b));
}
</pre>
4	UML включает: Dataset (ассоциация с Preprocessor), Preprocessor (агрегация с Model), абстрактный Model (наследование ClassificationModel, RegressionModel), Validator (зависимость от Model).
5	<pre>python
from sklearn.metrics import roc_auc_score, precision_recall_curve
auc = roc_auc_score(y_test, y_pred_proba)
precision, recall, _ = precision_recall_curve(y_test, y_pred_proba)
plt.plot(recall, precision); plt.savefig('pr_curve.png')
</pre>
6	<pre>``rust use nalgebra::{DMatrix, DVector, SVD}; pub struct LinearRegression { coef: DVector<f64> } impl LinearRegression { pub fn fit(X: &DMatrix<f64>, y: &DVector<f64>) -> Result<Self, &'static str> { let svd = SVD::new(X.clone(), true, true); let coef = svd.solve(y, 1e-12).map_err(_ "Matrix is singular"?); Ok(LinearRegression { coef }) } } ``</pre>
7	<pre>go
type BallTree struct {
 centroid []float64
 radius float64
 left, right *BallTree
 points [][]float64
}
func (bt *BallTree) Search(query []float64, k int) [][]float64 {
 // Рекурсивный поиск с отсечением по расстоянию до centroid + radius
}
</pre>
8	График покажет: ассигасу обычно максимальна при средних k (3-7), при больших k снижается. После PCA ассигасу может упасть, но время работы сократится в разы.

№ вопроса	Верный ответ
9	python from sklearn.pipeline import Pipeline from sklearn.preprocessing import StandardScaler class VarianceThresholdSelector(BaseEstimator, TransformerMixin): def __init__(self, threshold=1.0): self.threshold = threshold def fit(self, X, y=None): self.var_ = np.var(X, axis=0); return self def transform(self, X): return X[:, self.var_ > self.threshold] pipeline = Pipeline([('selector', VarianceThresholdSelector()), ('scaler', StandardScaler())])
10	julia using LinearAlgebra function ridge_regression(X, y, lambda) I = Matrix{Float64}(I, size(X,2), size(X,2)) beta = (X'X + lambda*I) \ (X'y) return beta end Время аналитического решения $\sim O(n^3)$, градиентный спуск может быть быстрее для очень больших n.
11	python def mape(y_true, y_pred): """ >>> mape([100, 200], [110, 190]) 5.0 """ return np.mean(np.abs((y_true - y_pred)/y_true))*100
12	r library(MASS) models <- list(lm(y~x, data), lm(y~poly(x,2), data), rpart(y~x, data)) <- sapply(models, function(m) c(R2=summary(m)\$r.squared, AdjR2=summary(m)\$adj.r.squared, BIC=BIC(m)))
13	scala import org.apache.spark.ml.clustering.KMeans val kmeans = new KMeans().setK(3).setSeed(1L) val model = kmeans.fit(df) val predictions = model.transform(df) predictions.write.parquet("hdfs://path/output")
14	python from sklearn.neighbors import kneighbors_graph from scipy.sparse.csgraph import minimum_spanning_tree import hdbscan clusterer = hdbscan.HDBSCAN(min_cluster_size=5) labels = clusterer.fit_predict(X)
15	c double manhattan_distance(double *a, double *b, int dim) { double sum = 0.0; for(int i=0; i<dim; i++) sum += fabs(a[i] - b[i]); return sum; } // K-medians: на каждой итерации вычисляем медианы по координатам
16	python import graphviz def plot_tree(tree, feature_names): dot_data = export_graphviz(tree, out_file=None, feature_names=feature_names) graph = graphviz.Source(dot_data) graph.render("decision tree")
17	Bayesian Optimization находит оптимальные параметры за меньшее число итераций (обычно 30-50) по сравнению с GridSearch (сотни), давая сравнимую или лучшую accuracy.
18	python from flask import Flask, request, jsonify import pickle app = Flask(__name__) model = pickle.load(open('model.pkl', 'rb')) @app.route('/predict', methods=['POST']) def predict(): data = request.get_json() prediction = model.predict([data['features']])[0] return jsonify({'decision': 'approved' if prediction==1 else 'denied'})
19	matlab load mnist_38.mat; % Загрузка данных для цифр 3 и 8 SVMModel = fitsvm(X_train, y_train, 'KernelFunction','rbf', 'BoxConstraint',1); predictions = predict(SVMModel, X_test); accuracy = sum(predictions==y_test)/numel(y_test);
20	c++ #include <dlib/svm.h> typedef dlib::polynomial_kernel<sample_type> kernel_type; dlib::svm_c_linear_trainer<kernel_type> trainer; trainer.set_kernel(kernel_type(3)); decision_function<kernel_type> df = trainer.train(samples, labels); dlib::serialize("model.dat") << df;

1. Напишите на языке Java два метода в классе `MLAlgorithms`: `knnClassify()` для классификации и `knnRegress()` для регрессии. Методы должны принимать массивы данных, параметр `k` и возвращать результат. Объясните, как будет различаться работа с типами возвращаемых данных (`int` vs `double`).

2. Разработайте программу на C#, которая считывает CSV-файл, определяет типы столбцов (числовой/категориальный), автоматически обрабатывает выбросы в числовых данных (используя межквартильный размах) и кодирует категориальные признаки методом Label Encoding.

3. Реализуйте алгоритм k-NN на языке C++ с использованием STL контейнеров (`vector`, `priority_queue`). Реализуйте две метрики расстояния:

Евклидову и Косинусную. Измерьте время выполнения для разных k (1, 5, 10) с помощью `chrono`.

4. Создайте UML-диаграмму классов для системы машинного обучения, включающей классы `Dataset`, `Preprocessor`, `Model` (с наследниками `ClassificationModel`, `RegressionModel`), `Validator`. Покажите связи между ними.
5. Напишите на Python программу, которая генерирует синтетические данные для бинарной классификации (`make_moons`), разбивает их на `train/test`, обучает логистическую регрессию и выводит в консоль оценку ROC-AUC и сохраняет график `precision-recall curve` в файл.
6. Реализуйте на языке Rust структуру `LinearRegression` с методами `fit` (использующий сингулярное разложение SVD) и `predict`. Обеспечьте безопасную работу с памятью и обработку ошибок для вырожденных матриц.
7. Оптимизируйте поиск ближайших соседей, реализовав Ball Tree на языке Go. Напишите рекурсивную функцию построения дерева и метод поиска k ближайших соседей с отсечением ветвей по условию.
8. Протестируйте свою реализацию k -NN на C++ на наборе данных MNIST (784 признака). Постройте график зависимости accuracy от k (от 1 до 20) и от размерности данных (используя PCA для уменьшения размерности до 50, 100, 200). Сделайте выводы.
9. Разработайте на Python конвейер (`pipeline`), который автоматически определяет, нужно ли масштабировать признак (проверяя его дисперсию), и применяет либо `StandardScaler`, либо оставляет как есть. Протестируйте на датасете с признаками разного масштаба.
10. Реализуйте на Julia функцию `ridge_regression(X, y, lambda)`, которая решает задачу ридж-регрессии с регуляризацией L2 аналитически: $\beta = (X'X + \lambda * I) \setminus X'y$. Сравните с градиентным спуском по времени выполнения.
11. Создайте на Python модуль `regression_metrics.py`, который экспортирует функции для расчета MAPE (Mean Absolute Percentage Error), RMSE (Root Mean Square Error) и Symmetric Mean Absolute Percentage Error (sMAPE). Добавьте `doctest`-тесты для каждой функции.
12. Напишите программу на R, которая строит несколько регрессионных моделей (линейная, полиномиальная 2-й степени, дерево решений) на одном

датасете, вычисляет для каждой R^2 , скорректированный R^2 и BIC, и представляет результаты в виде сводной таблицы (dataframe).

13. Реализуйте на языке Scala алгоритм K-Means с использованием Apache Spark DataFrame API. Программа должна считывать данные из HDFS, выполнять кластеризацию и сохранять результаты с метками кластеров обратно в HDFS в формате Parquet.

14. Напишите на Python функцию `hdbscan_clustering()`, реализующую алгоритм HDBSCAN* (иерархический DBSCAN). Используйте библиотеку `scikit-learn` для построения минимального остовного дерева (MST) и извлечения кластеров.

15. Реализуйте алгоритм кластеризации методом k-medians (использующий расстояние Манхэттена) на языке C. Программа должна использовать только базовые библиотеки (`stdio`, `math`) и эффективно работать с большими массивами в памяти.

16. Напишите на Python рекурсивную функцию построения дерева решений с использованием критерия информационного выигрыша (`information gain`) для категориальных признаков. Визуализируйте дерево с помощью библиотеки `graphviz`.

17. Используя фреймворк XGBoost на Python, настройте гиперпараметры модели (`max_depth`, `learning_rate`, `n_estimators`) с помощью оптимизации Bayesian Optimization (библиотека `hyperopt`). Сравните результаты с подбором по сетке (`GridSearchCV`).

18. Создайте веб-приложение на Flask (Python), которое принимает через POST-запрос JSON с признаками кредитной заявки и возвращает решение (одобрено/отказано) на основе заранее обученной модели Random Forest. Добавьте `endpoint` для оценки важности признаков модели.

19. Реализуйте на языке MATLAB или Octave SVM с гауссовым ядром (RBF) для классификации изображений цифр (только классы 3 и 8 из MNIST). Используйте встроенную функцию `quadprog` для решения задачи квадратичного программирования.

20. Напишите на C++ с использованием библиотеки `dlib` программу обучения SVM с полиномиальным ядром 3-й степени. Добавьте возможность сохранения обученной модели в файл и последующей загрузки для предсказания.