

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шитикова Александра Васильевна
Должность: директор Института агробиотехнологий
Дата подписания: 2025.08.11
Уникальный идентификатор документа:
fcd01ecb1fdf76848c517245ad12c3f716ce658



**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ
ФЕДЕРАЦИИ**

Государственное бюджетное образовательное учреждение высшего образования
**«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ –
МСХА имени К.А. ТИМИРЯЗЕВА»**
(ФГБОУ ВО РГАУ - МСХА имени К.А. Тимирязева)

Институт экономики и управления АПК
Кафедра статистики и кибернетики

УТВЕРЖДАЮ:

И.о. директора Института
агробиотехнологии

А.В. Шитикова

“ 28 ” 08

2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

**Б1.В.07 «Искусственный интеллект в гидрометеорологическом
обеспечении»**

для подготовки магистров

ФГОС ВО

Направление: 05.04.04 «Гидрометеорология»

Направленность: «Гидрометеорологическое обеспечение АПК»

Курс 1

Семестр 2

Форма обучения: очная

Год начала подготовки: 2025

Москва, 2025

Разработчики:

Быков Д.В., старший преподаватель

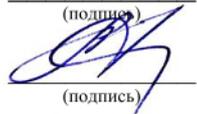
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

Уколова А.В., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

Рецензент:

Гавриловская Н.В., канд. техн. наук,

доцент

(ФИО, ученая степень, ученое звание)


(подпись)

«28» августа 2025 г.

Программа составлена в соответствии с требованиями ФГОС ВО, профессионального стандарта и учебного плана по направлению подготовки 05.04.04 Гидрометеорология

Программа обсуждена на заседании кафедры статистики и кибернетики. Протокол № 11 от «26» августа 2025 г.

И. о. заведующего кафедрой статистики и кибернетики

Уколова А.В., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)


(подпись)

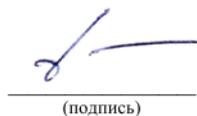
«26» августа 2025 г.

Согласовано:

Председатель учебно-методической комиссии института экономики и управления АПК

Гупалова Т.Н., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)


(подпись)

«28» августа 2025 г.

И. о. заведующего выпускающей кафедрой статистики и кибернетики

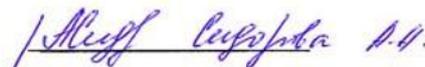
Уколова А.В., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

Заведующий отделом комплектования ЦНБ


(подпись)

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	4
1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	4
2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ	4
3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	5
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	8
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ ПО СЕМЕСТРАМ	8
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	8
4.3 ЛЕКЦИИ/ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	9
5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	12
6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	13
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ	13
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ	32
7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	34
7.1 ОСНОВНАЯ ЛИТЕРАТУРА	34
7.2 ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	34
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	35
9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ.....	35
10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ.....	36
11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ .	38
Виды и формы отработки пропущенных занятий	38
12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....	38

АННОТАЦИЯ

рабочей программы учебной дисциплины Б1.В.07 «Искусственный интеллект в гидрометеорологическом обеспечении» для подготовки магистров по направлению 05.04.04 «Гидрометеорология» направленности «Гидрометеорологическое обеспечение АПК»

Цель освоения дисциплины. Основная цель дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении» – овладение студентами основными методами искусственного интеллекта, приобретение навыков по разработке моделей искусственных нейронных сетей, изучение основных подходов к обучению и применению искусственных нейронных сетей для решений задач анализа и обработки данных в гидрометеорологии.

Место дисциплины в учебном плане: дисциплина включена в часть учебного плана по направлению подготовки 05.04.04 «Гидрометеорология», формируемую участниками образовательных отношений.

Требования к результатам освоения дисциплины: в результате освоения дисциплины формируются следующие компетенции (индикаторы): УК-4 (УК-4.3); ПКос-1 (ПКос-1.1; ПКос-1.2); ПКос-4 (ПКос-4.3).

Краткое содержание дисциплины:

Основы языка программирования Python. Объекты. Типы объектов. Переменные. Операции. Строка. Подключение модулей. Список. Кортеж. Итерация по списку. Условный оператор. Циклы while, for в Python. Функции в Python. Классы в Python.

Разработка и обучение моделей многослойного персептрона с использованием современных библиотек и фреймворков. Подбор оптимальных гиперпараметров для моделей многослойного персептрона при решении задачи регрессии. Разработка рекуррентной нейронной сети (RNN). Разработка сети долгой краткосрочной памяти (LSTM).

Общая трудоемкость дисциплины составляет 3 зачетные единицы (108 часов).

Промежуточный контроль: зачет.

1. Цель освоения дисциплины

Целью освоения дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении» является овладение студентами основными методами искусственного интеллекта, приобретение навыков по разработке моделей искусственных нейронных сетей, изучение основных подходов к обучению и применению искусственных нейронных сетей для решений задач анализа и обработки данных в гидрометеорологии.

2. Место дисциплины в учебном процессе

Дисциплина «Искусственный интеллект в гидрометеорологическом обеспечении» относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (модули)» учебного плана. Дисциплина «Искусственный интеллект в гидрометеорологическом обеспечении»

реализуется в соответствии с требованиями ФГОС, ОПОП ВО и Учебного плана по направлению 05.04.04 «Гидрометеорология».

Предшествующими курсами, на которых непосредственно базируется дисциплина «Искусственный интеллект в гидрометеорологическом обеспечении» являются «Гидрометеорологические информационные системы», «Современные проблемы гидрометеорологии».

Дисциплина «Искусственный интеллект в гидрометеорологическом обеспечении» является основополагающей для изучения следующих дисциплин: «Климатические прогнозы и прогнозы погоды», «Современные методы агрометеорологических наблюдений и анализа», а также для подготовки выпускной квалификационной работы.

Особенностью дисциплины является изучение инструментов создания искусственного интеллекта, разработка и программирование моделей искусственного интеллекта, в том числе для решения задач в области анализа данных.

Рабочая программа дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

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

Образовательные результаты освоения дисциплины обучающимся, представлены в таблице 1.

Таблица 1

Требования к результатам освоения учебной дисциплины

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
1.	УК-4	Способен применять современные коммуникативные технологии, в том числе на иностранном(ых) языке(ах), для академического и профессионального взаимодействия	УК-4.3 Аргументированно и конструктивно отстаивает свои позиции и идеи в академических и профессиональных дискуссиях, в том числе на иностранном языке		Аргументированно и конструктивно отстаивать свои позиции и идеи в академических и профессиональных дискуссиях, в том числе на иностранном языке	
2.	ПКос-1	Способен использовать современные методы наблюдений, обработки и интерпретации информации при проведении научных и производственных исследований в растениеводстве, имеющих гидрометеорологическую направленность, с применением цифровых технологий, в том числе на основе искусственного интеллекта	ПКос-1.1 Знает современные методы наблюдений, в том числе с применением цифровых технологий и платформенных решений, статистической обработки и интерпретации результатов научных и производственных наблюдений гидрометеорологических процессов и рядов, формулирования выводов	Современные методы наблюдений, в том числе с применением цифровых технологий и платформенных решений, статистической обработки и интерпретации результатов научных и производственных наблюдений гидрометеорологических процессов и рядов,		

				формулирования выводов		
			ПКос-1.2 Умеет использовать современные цифровые технологии, компьютерную графику и текстовые процессоры, при подготовке отчетов о проведенных научных и производственных исследованиях		Использовать современные цифровые технологии, компьютерную графику и текстовые процессоры, при подготовке отчетов о проведенных научных и производственных исследованиях	
3.	ПКос-4	Способен осуществлять гидрометеорологическое обеспечение и экологическую экспертизу при планировании, организации и строительстве хозяйственных объектов АПК, соблюдением мер климатической безопасности	ПКос-4.3 Владеет статистическими методами исследований, прогнозирования и оценки экологической безопасности при планировании, организации и строительстве хозяйственных объектов, а также способами адаптации к неблагоприятным изменениям климата			Статистическими методами исследований, прогнозирования и оценки экологической безопасности при планировании, организации и строительстве хозяйственных объектов, а также способами адаптации к неблагоприятным изменениям климата

4. Структура и содержание дисциплины

4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 3 зачетные единицы (108 часов), их распределение по видам работ и по семестрам представлено в таблице 2.

ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 2а

Распределение трудоёмкости дисциплины по видам работ по семестрам

Вид учебной работы	Трудоёмкость (семестр № 2)/*
	час.
Общая трудоёмкость дисциплины по учебному плану	108/4
1. Контактная работа	24,25/4
Аудиторная работа	24,25/4
<i>лекции (Л)</i>	12
<i>практические занятия (ПЗ)</i>	12/4
<i>контактная работа на промежуточном контроле (КРА)</i>	0,25
2. Самостоятельная работа (СРС)	83,75
<i>самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к практическим занятиям)</i>	74,75
<i>подготовка к зачету</i>	9
Вид промежуточного контроля:	Зачет

* в том числе практическая подготовка

4.2 Содержание дисциплины

Таблица 3

Тематический план учебной дисциплины

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ/*	ПКР	
Раздел 1. Основы программирования на Python	46,75/4	6	6/4	-	34,75
Раздел 2. Разработка моделей искусственных нейронных сетей	52	6	6	-	40
<i>контактная работа на промежуточном контроле (КРА)</i>	0,25	-	-	0,25	-
<i>подготовка к зачету (контроль)</i>	9	-	-	-	9
Итого по дисциплине	108/4	12	12/4	0,25	83,75

* в том числе практическая подготовка

Раздел 1. Основы программирования на Python

Тема 1. Базовые элементы языка программирования Python

Основы языка программирования Python. Объекты. Типы объектов. Переменные. Операции. Строка. Подключение модулей. Список. Кортеж. Итерация по списку. Условный оператор. Циклы while, for в Python. Функции в Python. Классы в Python.

Раздел 2. Разработка моделей искусственных нейронных сетей

Тема 1. Программирование искусственных нейронных сетей

Разработка и обучение моделей многослойного персептрона с использованием современных библиотек и фреймворков. Подбор оптимальных гиперпараметров для моделей многослойного персептрона при решении задачи регрессии. Разработка рекуррентной нейронной сети (RNN). Разработка сети долгой краткосрочной памяти (LSTM).

4.3 Лекции/практические занятия

Таблица 4

Содержание лекций/практических занятий и контрольные мероприятия

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия	Кол-во часов/ из них практическая подготовка
1.	Раздел 1. Основы программирования на Python		УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3		12/4
	Тема 1. Базовые элементы языка программирования Python	Лекция № 1. Основы языка программирования Python. Объекты. Типы объектов. Переменные. Операции.	ПКос-1.1		2
		Практическая работа № 1. Объекты. Типы объектов. Переменные. Операции в Python.	УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3	устный опрос, защита практической работы	2/2
		Лекция № 2. Основы языка программирования Python. Строка. Подключение модулей. Список. Кортеж. Итерация по списку. Условный оператор.	ПКос-1.1		2
		Практическая работа № 2. Строка. Подключение модулей. Список. Кортеж. Итерация по списку. Условный оператор в Python.	УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3	устный опрос, защита практической работы	2/2

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия	Кол-во часов/ из них практическая подготовка
		Лекция № 3. Основы языка программирования Python. Циклы while, for. Функции.	ПКос-1.1		1
		Практическая работа № 3. Циклы while, for в Python. Функции в Python.	УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3	устный опрос, защита практической работы	1
		Лекция № 4. Основы языка программирования Python. Классы.	ПКос-1.1		1
		Практическая работа № 4. Классы в Python.	УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3	устный опрос, защита практической работы	1
2.	Раздел 2. Разработка моделей искусственных нейронных сетей		УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3		12
	Тема 1. Программирование искусственных нейронных сетей	Лекция № 5. Инструменты разработки и обучения различных архитектур искусственных нейронных сетей.	ПКос-1.1		2
		Практическая работа № 5. Разработка и обучение моделей многослойного персептрона с использованием современных библиотек и фреймворков.	УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3	устный опрос, защита практической работы	2
		Лекция № 6. Особенности решения задачи регрессии на основе многослойного персептрона.	ПКос-1.1		2
		Практическая работа № 6. Подбор оптимальных гиперпараметров для моделей многослойного персептрона при решении задачи регрессии.	УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3	устный опрос, защита практической работы	2
		Лекция № 7. Рекуррентные нейронные сети (RNN).	ПКос-1.1		1
		Практическая работа № 7. Разработка рекуррентной нейронной сети (RNN).	УК-4.3;		устный опрос, защита

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия	Кол-во часов/ из них практическая подготовка
			ПКос-1.1; ПКос-1.2; ПКос-4.3	практической работы	
		Лекция № 8. Сети долгой краткосрочной памяти (LSTM).	ПКос-1.1		1
		Практическая работа № 8. Разработка сети долгой краткосрочной памяти (LSTM).	УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3	устный опрос, защита практической работы	1

Таблица 5

Перечень вопросов для самостоятельного изучения дисциплины

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
Раздел 1. Основы программирования на Python		
1.	Тема 1. Базовые элементы языка программирования Python	1. Преимущества языка программирования Python при создании систем искусственного интеллекта. (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3) 2. Библиотеки языка программирования Python для построения моделей искусственных нейронных сетей. (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3)
Раздел 2. Разработка моделей искусственных нейронных сетей		
2.	Тема 1. Программирование моделей искусственных нейронных сетей	1. Современные подходы к параметрически-эффективному обучению (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3). 2. Алгоритмы оптимизации с адаптивной скоростью обучения (AdaGrad, RMSProp, Adam), применяемые при обучении искусственных нейронных сетей (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3). 3. Алгоритмы обработки данных, используемые при обучении моделей искусственных нейронных сетей; алгоритмы автоматизации подбора архитектур (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3). 4. Современные модели искусственных нейронных сетей, используемых для прогнозной и прескриптивной аналитики в агропромышленном комплексе (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3). 5. Влияние когнитивных искажений и предвзятостей на принимаемые решения с использованием моделей искусственных нейронных сетей; оценка надёжности данных, получаемых от моделей искусственных нейронных сетей на основе контекста, источников, методики и возможных рисков (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3). 6. Оценка целесообразности и ограничений применения моделей искусственных нейронных сетей для различных задач с

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
		<p>учётom технических, социальных и правовых условий (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3).</p> <p>7. Задачи, решаемые с использованием самоорганизующихся карт Кохонена (SOM) (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3).</p> <p>8. Особенности реализации самоорганизующихся карт Кохонена (SOM). (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3)</p> <p>9. Задачи, решаемые с использованием RBF-сети (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3).</p> <p>10. Особенности реализации RBF-сети. (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3).</p> <p>11. Задачи, решаемые с использованием ограниченной машины Больцмана (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3).</p> <p>12. Особенности реализации ограниченной машины Больцмана (УК-4.3; ПКос-1.1; ПКос-1.2; ПКос-4.3).</p>

5. Образовательные технологии

Таблица 6

Применение активных и интерактивных образовательных технологий

№ п/п	Тема и форма занятия	Наименование используемых активных и интерактивных образовательных технологий (форм обучения)
1.	Практическая работа № 1. Объекты. Типы объектов. Переменные. Операции в Python.	ПЗ Компьютерная симуляция
2.	Практическая работа № 2. Строка. Подключение модулей. Список. Кортеж. Итерация по списку. Условный оператор в Python.	ПЗ Компьютерная симуляция
3.	Практическая работа № 3. Циклы while, for в Python. Функции в Python.	ПЗ Компьютерная симуляция
4.	Практическая работа № 4. Классы в Python.	ПЗ Компьютерная симуляция
5.	Практическая работа № 5. Разработка и обучение моделей многослойного персептрона с использованием современных библиотек и фреймворков.	ПЗ Компьютерная симуляция
6.	Практическая работа № 6. Подбор оптимальных гиперпараметров для моделей многослойного персептрона при решении задачи регрессии.	ПЗ Компьютерная симуляция
7.	Практическая работа № 7. Разработка рекуррентной нейронной сети (RNN).	ПЗ Компьютерная симуляция
8.	Практическая работа № 8. Разработка сети долгой краткосрочной памяти (LSTM).	ПЗ Компьютерная симуляция

6. Текущий контроль успеваемости и промежуточная аттестация по итогам освоения дисциплины

6.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности

1) Вопросы для защиты практических работ

1. Дайте определение понятию «объект» в контексте языка Python.
2. Объясните значение характеристики «динамически типизируемый» относительно языка программирования.
3. Перечислите названия нескольких встроенных типов объектов. Приведите примеры объектов таких типов.
4. Что из себя представляет преобразование типов?
5. Дайте определение понятию «переменная».
6. Каким образом происходит создание переменной?
7. Какие существуют правила при выборе имени переменной?
8. Дайте определение понятиям «оператор», «операнд», «операция».
9. Что подразумевает приоритет выполнения операций?
10. Что из себя представляет встроенные функции Python?
11. Перечислите названия нескольких встроенных функций Python. Приведите примеры их использования.
12. Дайте определение понятию «строка».
13. Что из себя представляют и для чего используются управляющие последовательности (последовательности экранирования) строки?
14. Что подразумевается под индексацией элементов строки?
15. Дайте определение понятиям «срез», «нарезание» относительно строки.
16. Перечислите несколько методов строк. Объясните принцип их функционирования.
17. Дайте определение понятию «модуль».
18. Каким образом происходит подключение модуля и обращение к элементам модуля?
19. Перечислите несколько элементов модуля math. Объясните их назначение.
20. Дайте определение понятию «список».
21. Каким образом можно создать пустой список?
22. Каким образом можно создать список из нескольких элементов?
23. Каким образом можно в пустой список добавить несколько элементов?
24. Что из себя представляет конкатенация списков?
25. Что из себя представляет операция «повторение» относительно списка?
26. Каким образом можно осуществить проход по элементам списка?
27. Что из себя представляет условный оператор if?
28. Что из себя представляет конструкция «if-elif-else»?
29. Что из себя представляет блок инструкций в Python? Для чего используется табуляция в блоке инструкций?

30. Каким образом можно проверить, является ли список пустым, с помощью оператор `if`?
31. Дайте определение понятию «цикл».
32. Что из себя представляет цикл `while`?
33. Приведите простую форму цикла `while`, включающую только обязательные части.
34. Для чего используется конструкция `else` цикла?
35. Для чего используется оператор `break` в цикле?
36. Для чего используется оператор `continue` в цикле?
37. Для чего может использоваться оператор `pass` в цикле?
38. Приведите полную форму цикла `while`.
39. В каком случае цикл `while` будет выполняться «бесконечно»?
40. Что из себя представляет цикл `for`?
41. В каких случаях стоит использовать цикл `while`, а в каких – цикл `for`?
42. Дайте определение понятию «функция» относительно языка программирования.
43. Каким образом происходит создание функции?
44. Приведите общую форму записи для создания функции.
45. Что подразумевается под объектно-ориентированным программированием?
46. Дайте определение понятию «класс».
47. Дайте определение понятию «объект».
48. Дайте определение понятию «наследование».
49. Дайте определение понятию «инкапсуляция».
50. Дайте определение понятию «полиморфизм».
51. Каким образом происходит определение класса? Приведите общую форму записи при определении класса.
52. Каким образом происходит определение экземпляра (объекта) класса?
53. Что подразумевается под атрибутом класса?
54. Что подразумевается под методом класса?
55. Есть ли разница между атрибутом класса и атрибутом объекта?
56. Для чего используется ключевое слово `self`?
57. Для чего используется функция `super()`?
58. Библиотеки и фреймворки, используемые при разработке и обучении моделей многослойного персептрона.
59. Возможности библиотеки TensorFlow, используемой при разработке и обучении моделей многослойного персептрона.
60. Возможности библиотеки PyTorch, используемой при разработке и обучении моделей многослойного персептрона.
61. Основные операции с матрицами, используемые при разработке многослойного персептрона.
62. Персептрон.
63. Многослойный персептрон.
64. Гиперпараметры модели многослойного персептрона.
65. Весовые коэффициенты модели многослойного персептрона.
66. Способы формирования начальных значений весов нейронов.

67. Скорость (темп) обучения модели многослойного персептрона.
68. Функции потерь.
69. Сущность процесса обучения модели многослойного персептрона.
70. Алгоритм обратного распространения ошибки.
71. Метод градиентного спуска.
72. Тестирование модели многослойного персептрона.
73. Изменение конфигурации модели многослойного персептрона.
74. Сохранение обученной модели многослойного персептрона.
75. Загрузка и применение сохраненной обученной модели многослойного персептрона.
76. Радиально-базисная сеть (RBF-сеть): сущность, основные элементы.
77. Радиально-базисная сеть (RBF-сеть): решаемые задачи.
78. Радиально-базисная сеть (RBF-сеть): особенности разработки модели.
79. Радиально-базисная сеть (RBF-сеть): особенности обучения модели.
80. Сеть долгой краткосрочной памяти (LSTM): сущность, основные элементы.
81. Сеть долгой краткосрочной памяти (LSTM): решаемые задачи.
82. Сеть долгой краткосрочной памяти (LSTM): особенности разработки модели.
83. Сеть долгой краткосрочной памяти (LSTM): особенности обучения модели.

2) Перечень вопросов, выносимых на промежуточную аттестацию (зачет)

1. Сущность понятий «искусственный нейрон», «искусственная нейронная сеть».
2. Гиперпараметры модели искусственной нейронной сети.
3. Весовые коэффициенты модели искусственной нейронной сети.
4. Способы формирования начальных значений весов искусственной нейронной сети.
5. Скорость (темп) обучения модели искусственной нейронной сети.
6. Функции потерь модели искусственной нейронной сети.
7. Сущность процесса обучения модели искусственной нейронной сети.
8. Алгоритм обратного распространения ошибки.
9. Метод градиентного спуска.
10. Перечислите и охарактеризуйте современные библиотеки и фреймворки для разработки моделей искусственных нейронных сетей.
11. Перечислите и охарактеризуйте основные архитектуры неглубоких искусственных нейронных сетей.
12. Перечислите и охарактеризуйте основные архитектуры глубоких искусственных нейронных сетей.
13. Многослойный персептрон (MLP): сущность, основные элементы, решаемые задачи.

14. Многослойный перцептрон (MLP): особенности функционирования, обучения и программной реализации.
15. Рекуррентная нейронная сеть (RNN): сущность, основные элементы, решаемые задачи.
16. Рекуррентная нейронная сеть (RNN): особенности функционирования, обучения и программной реализации.
17. Сеть долгой краткосрочной памяти (LSTM): сущность, основные элементы, решаемые задачи.
18. Сеть долгой краткосрочной памяти (LSTM): особенности функционирования, обучения и программной реализации.

Пример работ

Практическая работа № 1.

Объекты. Типы объектов. Переменные. Операции в Python

Задача № 1. Объекты. Типы объектов. Переменные. Арифметические и логические операции

1. Создайте объект с именем *a* и значением 10.
2. Создайте объект *b* со значением 15.
3. Выведите в консоль значения объектов *a*, *b*.
4. Определите тип объектов *a*, *b*.
5. Измените переменную *a* так, чтобы она ссылалась на объект со значением 8 (объект *a* должен быть со значением 8).
6. Увеличьте значение объекта *a* на 3.
7. Увеличьте значение объекта *a* в 5 раз.
8. Объект *c* определите как сумму *a* и *b*.
9. Определите истинность высказывания: *c* равно *a*.
10. Определите истинность высказывания: *c* не равно 0.
11. Определите, является ли *c* больше, чем *a*.
12. Определите, истинно ли то, что *b* меньше или равно *a*.
13. Определите истинность высказывания: объекты *a*, *b*, *c* больше 10.
14. Определите истинность высказывания: хотя бы один из объектов *a*, *b*, *c* больше 10.
15. Объект *h* определите как a^b .
16. Объект *t* определите как следующий текст «Python language».
17. Выведите в консоль значения объектов *a*, *b*, *c*, *h*, *t*.
18. Определите тип объектов *a*, *b*, *c*, *h*, *t*.
19. Измените тип объекта *a* на float путем явного преобразования типов.
20. Измените значение и тип объекта *a* на float путем неявного преобразования типов.
21. Объект *b* уменьшите на 3,5.
22. Выведите в консоль значения объектов *a*, *b*, *c*, *h*, *t*.
23. Определите модуль *b*.

24. Найдите целую часть от деления b на a .

25. Найдите остаток от деления b на c и округлите до целых.

Задача № 2. Встроенные функции

1. Создайте объект `var_in`, значение которого необходимо ввести в консоль.
2. Определите максимальное значение из объектов a, b, c, h, var_in .
3. Определите минимальное значение из объектов a, b, c, h, var_in .
4. Преобразуйте переменные в двоичное представление.
5. Преобразуйте переменные в восьмеричное представление.
6. Преобразуйте переменные в шестнадцатеричное представление.
7. Создайте объект r со значением 18,947301. Округлите значение до тысячных.
8. Создайте объект `r_in`, значение которого формируется следующим образом. В консоль необходимо ввести два числа:
 - $a1$ – исходное значение в виде вещественного числа
 - $a2$ – значение в виде целого числа, отражающее количество знаков после запятой.

В переменную `r_in` сохранятся результат округления числа $a1$ до $a2$ знаков после запятой.

Пример:

In: Введите два числа: 5,40917 2

Out: r_in = 5,41

Задача № 3. Очередность выполнения операций

Условие: имеется некоторый алгоритм – последовательность простых (состоящих из одного оператора и двух операндов) математических выражений для расчета определенного числового значения. При этом на первом шаге алгоритма оба операнда являются исходными числами, на втором шаге – первый операнд является результатом выражения предыдущего (первого) шага и т.д.

Требуется: составить смешанное выражение на языке Python (состоящее из нескольких подвыражений) для расчета итогового числового значения с учетом последовательности выполнимых действий (математических операций).

Ответ предоставьте в двух вариантах:

- а. используйте как можно больше круглых скобок;
- б. используйте как можно меньше круглых скобок.

Пример:

Имеется следующий алгоритм из пяти шагов:

- 1) $7 + 3$
- 2) $* 10$
- 3) $- 4$
- 4) -90
- 5) $** 2$

Описание принципа работы алгоритма:

На шаге 1 выражение примет вид $7 + 3$, результат выражения = 10.

На шаге 2 выражение примет вид $10 * 10$, результат = 100.

На шаге 3 выражение примет вид $100 - 4$, результат = 96.

На шаге 4 выражение примет вид $96 - 90$, результат = 6.

На шаге 5 выражение примет вид $6 ** 2$, результат = 36.

Выражения на языке Python для правильного расчета результата:

a. $((((7 + 3) * 10) - 4) - 90) ** 2$

b. $((((7 + 3) * 10 - 4) - 90) ** 2$

Решите задачу № 3 для следующих алгоритмов:

Алгоритм 1	Алгоритм 2	Алгоритм 3
1) $4 + 3$	1) $2 * 3$	1) $1 / 10$
2) $- 11$	2) $+ 10$	2) $+ 4.1$
3) $* 2$	3) $* 2$	3) $- 8$
	4) $** 2$	4) $- 2$
		5) $** 2$
		6) $* 10$

Задача № 4

1. Разработайте программный код, с помощью которого можно будет вводить номер товара в виде какого-то числа a и получать в консоли ответ в виде текстового сообщения «Товар номер a поступил на склад».

Пример:

In: Введите номер товара: 7

Out: Товар номер 7 поступил на склад

2. Для п. 1 добавьте возможность вводить помимо номера товара вес этого товара в граммах.

Пример:

In: Введите номер товара и его вес в граммах (через пробел): 7 35.11

Out: Товар номер 7 весом 35.11 г поступил на склад

3. Для п. 2 добавьте расчет веса в кг.

Пример:

In: Введите номер товара и его вес в граммах (через пробел): 7 35.11

Out: Товар номер 7 весом 35.11 г (0.03511 кг) поступил на склад

4. Для п. 3 добавьте возможность вводить данные по двум товарам. В консоль отобразите информацию об итоговом количестве товаров, общем весе всех товаров.

Пример:

Out: Введите информацию по 2 товарам:

In: Введите номер товара и его вес в граммах (через пробел): 7 35.11

Out: Товар номер 7 весом 35.11 г (0.03511 кг) поступил на склад

In: Введите номер товара и его вес в граммах (через пробел): 19 1307.84

Out: Товар номер 19 весом 1307.84 г (1.30784 кг) поступил на склад

Out: На складе 2 товара (с номерами 7, 19) с общим весом 1342.95 г (1.34295 кг)

Практическая работа № 2.

Строка. Подключение модулей. Список. Кортеж. Итерация по списку. Условный оператор в Python

Часть № 1. Строка. Подключение модулей.

Задача № 1

Создайте объект *name*: « My name is »

1. Уберите лишние пробелы.
2. Выведите в консоль текст: «My name is ФИО» (ФИО – ваше собственное).
3. Определите число символов в п.2.
4. Выведите в консоль текст:
«My name is
ФИО»
5. Выведите в консоль все слова п.2 через знак табуляции.
6. Для п.5 уберите все символы экранирования.
7. Повторите 8 раз фразу из п.2.
8. Выведите в консоль «MY NAME IS» и проверьте, все ли символы заглавные.
9. Выведите все символы в п.2 с заглавных букв.
10. Выведите в консоль «NAME» как часть строки из п.9.
11. Проверьте, есть ли символы «m», «i», «t» в п.2.
12. Замените «na» на «****» в п.2.
13. Найдите позицию символа «u» в п.2.

Задача № 2

Создайте объект *t*: «a1 a2 a3».

1. Разделите строку на подстроки и добавьте их в список.
2. Разделите объект *t* на 2 объекта.
3. Соедините элементы списка из п.1 в одну строку через знак «,».
4. Разбейте объект в п.3 на 2 подстроки через знак «,», при этом поиск знака осуществите справа налево.

Задача № 3

Создайте переменную *x*, ссылающуюся на значение, соответствующее вашему варианту. Например, если вариант 1, то $x = 1$.

Рассчитайте результат применения следующих формул:

1. $x + 10!$
2. $(x - x^2 - x^3) + (x + x^2 + x^3)$
3. $\sqrt{|e^x + x! \cdot \pi|}$, ответ округлите в меньшую сторону;
4. $(\lg(x) + \sqrt{x^3})^{x-5}$, ответ округлите в большую сторону;
5. $x^e + \frac{\sin(x \cdot \pi)}{\tan x^{-1}}$

Задача № 4

Создайте переменную x , ссылающуюся на значение, соответствующее вашему варианту. Например, если вариант 1, то $x = 1$.

Вывести на печать, заменив «*» фактическими значениями:

1. «Наибольший общий делитель чисел x , 20, 50 равен *»
2. «Наибольший общий делитель чисел $10x$, 20, 50, 100, 15000 равен *»
3. «Наименьшее общее кратное чисел x , 18, 50 равно *»
4. « $\cos(x) = *.***$ » (округлить до 3 знаков после запятой)
5. « $\sin^2(x) + \cos^2(x) = **. ** = 1$ » (округлить до 2 знаков после запятой)

Часть № 2. Список. Кортеж. Итерация по списку. Условный оператор в Python

Задача № 1. Список

Создайте список *list1*: 2, 3, 4, 2, 3, 2, 3, 4, 1, 1, 1, 5, 6, 6, 6.

1. Добавить в конец списка элементы 5, 7, 8.
2. Отсортировать список по возрастанию.
3. Выбрать элементы с 5 по 8 включительно.
4. Выбрать элементы с 3 по 10 не включительно.
5. Выбрать 3, 5 и 12 элементы.
6. Удалить элемент, стоящий на 4 месте в отсортированном списке.
7. На вторую позицию вставить значение 6.
8. Проверить есть ли значение 9 и 1 в списке.
9. Создать список *list2* как копию списка *list1*.
10. Выстроить элементы списка *list2* в обратном порядке.
11. Удалить из *list2* последний элемент.
12. Добавить на 5 позицию значение 12 в список *list2*.
13. Удалить 5 и 3 элементы из *list2*.
14. Объединить списки *list1* и *list2* и вывести их 3 раза в консоль.
15. Определить, сколько раз в списке *list2* встречается значение 6.
16. Преобразовать список *list2* в строку, предусмотрев следующие разделители: «,», «_», «/».
17. Найти корень из каждого значения списка *list2*.
18. Найти максимальное и минимальное значение списка *list2*.
19. Создать список *list3* от 3 до 15.
20. Создать список *list4* от 20 до 3.

Задача № 2. Кортеж

1. Создайте кортеж *c*: 4, 3, 11, 7, 2, 1, 100, 2, 4, 2.
2. Попробуйте добавить новый элемент в *c*, удалить элемент из кортежа *c*, изменить элемент с индексом 3.
3. Создайте кортеж, содержащий элементы от значения 10 до значения 555 с шагом 7.
4. Определите, сколько раз встречается «2» в кортеже *c*, используя при этом только специальный метод.
5. Определите, сколько раз встречается «17» в кортеже *c*, используя при этом только специальный метод.

6. Определите индекс элемента «100» в кортеже *c*, используя при этом только специальный метод.
7. Создайте список и преобразуйте его в кортеж.
8. Преобразуйте кортеж *c* в список для его изменения, а затем обратно в кортеж. Используя функцию `list()`, добавьте новый элемент в *c*, удалите элемент из кортежа *c*, измените элемент с индексом 3. Объект *c* после выполнения пункта 7 должен быть кортежем.

Задача № 3. Условный оператор

1. Ввести любое число с клавиатуры. Составить программу, которая определяет принадлежность введенного числа к интервалу (-10, 10).
2. Ввести любые 3 числа с клавиатуры. Составить программу, которая определяет, какое из трех чисел наибольшее.
3. Ввести любое число с клавиатуры. Определить четное данное число или нечетное.
4. Ввести любые 2 числа с клавиатуры. Вычесть от большего меньшее и результат вывести на экран.

Задача № 4

Напишите программу, которая будет выводить все нечетные числа из диапазона от 39 до 248 и остановится, если встретится 139.

Задача № 5

Дан список *list5* = [11, 5, 8, 32, 15, 3, 20, 132, 21, 4, 555, 9, 20]. Необходимо вывести элементы, которые одновременно меньше 30 и делятся на 3 без остатка. Все остальные элементы списка необходимо просуммировать и вывести конечный результат.

Задача № 6

Выведите все числа от 0 до N, где

1. N = 66.
2. N = -31.

Задача № 7

Вывести на экран циклом пять строк из нулей, причем каждая строка должна быть пронумерована.

Задача № 8

Найти сумму ряда чисел от 1 до 100. Полученный результат вывести на экран.

Задача № 9

Дано семь чисел. Найти количество положительных чисел среди них:

1. 5964, -12, -68874, 101, -103, -741, 36985.
2. -713, -12563, -89, -45698, -898, -75632, -635.

Задача № 10

Даны три числа. Вывести на экран «yes», если среди них есть одинаковые, иначе вывести «ERROR»:

1. 956820, 956620, 936820.
2. 24930566, 24960566, 24930566.
3. 3496, 3496, 3496.

Задача № 11

Вывести на экран все чётные целые числа в диапазоне от 1 до 698.

Задача № 12

Посчитать сумму числового ряда:

1. от 0 до 14 (пример: $0+1+2+3+\dots+14$).
2. от 569 до 601.
3. от -65 до 12.

Задача № 13

Перемножить все нечётные целые числа в диапазоне от 0 до 84.

Задача № 14

Записать в массив все целые числа в диапазоне от 54 до 3945, кратные 5.

Задача № 15

Даны три числа. Вывести на экран «yes», если можно взять какие-то два из них и в сумме получить третье:

1. 9760, 3594, 6166.
2. 56783, 49998, 6784.

Задача № 16

Напишите программу, которая будет по номеру месяца выводить время года. Например, если введено 2, то следует вывести «Зима».

Задача № 17

Условие: дан текст «Алгоритм (от лат. написания арабского имени аль-Хорезми – Algorithmi), инструкция, точное описание способа действия с использованием простых, общепонятных элементов (например, операций). В математике понятие алгоритма сужается и уточняется следующим образом. Действие состоит в последовательности переходов от одного состояния вычисления (процесса работы алгоритма) к другому; состояния – это конструктивные объекты (например, слова в данном алфавите; в частности, целые числа в десятичной или двоичной записи). Алгоритм также является конструктивным объектом. Первое состояние называется исходным данным, последнее – результатом работы алгоритма. Фиксированный алгоритм можно применять к различным исходным данным; для некоторых он может не заканчивать работу. Тем самым алгоритм задаёт (возможно, не всюду определённую) функцию, вычисляемую этим алгоритмом. Такие функции

называются вычислимыми. Понятия алгоритма и вычислимой функции относятся к исходным понятиям математики и через другие понятия не выражаются. Рассматриваются расширения понятия алгоритма, например вероятностные алгоритмы, т. н. алгоритмы с оракулом, алгоритмы взаимодействия с окружающей средой, параллельные алгоритмы. Часто алгоритм определяется с помощью абстрактной вычислительной машины, получающей на вход программу действия и исходное данное. До конца 19 в. алгоритм – общее понятие, относящееся к известным алгоритмам, таким как алгоритм выполнения арифметических операций в десятичной системе счисления, алгоритм дифференцирования функций, алгоритм Евклида нахождения общей меры отрезков или наибольшего общего делителя многочленов. В 1900–1910-х гг. были осознаны трудности в построении общего алгоритма решения некоторых массовых проблем. В 1930-е гг. предложены математические определения понятия вычислимой функции, исходящие из представлений о том, что может делать человек-вычислитель; среди них – понятие рекурсивной функции и понятие функции, вычислимой машиной Тьюринга. Тогда же была доказана эквивалентность различных понятий вычислимой функции и классов вычисляемых функций, порождаемых этими понятиями; сформулирован т. н. тезис Чёрча, принятый в качестве естественно-научного факта: класс вычисляемых функций совпадает с любым из упомянутых выше классов. Развитие компьютерных технологий не изменило представлений о классе функций, вычисляемых алгоритмами. Построение и анализ конкретных алгоритмов, предназначенных для выполнения компьютером, относится к программированию. Выделяются также классы вычислительных алгоритмов и обучающихся алгоритмов.» [Ошибка! Источник ссылки не найден].

Требуется:

1. Очистить представленный выше текст от знаков препинания.
2. Разбить текст на слова и создать список слов под именем L1 (допускается дублирование слов).
3. Найти индекс определенного слова.
4. Определить, сколько раз слово встречается в списке.
5. Добавить 5 слов в список (в конец).
6. Добавить 5 слов в список (в начало).
7. Сделать копию списка под именем L2.
8. Удалить из списка L2 каждое третье слово.
9. Удалить из списка L2 первое слово.
10. Удалить из списка L2 определенное слово.
11. Создать список L3 из списка L2, повторенного 4 раза.
12. Очистить список L2.
13. Добавить в список L2 несколько слов.
14. Удалить из списка L3 любые 10 слов, которые расположены подряд.

Задача № 18

1. Вывести каждое слово списка L1 в консоль.
2. Для п.1 добавить индекс слова, т.е. чтобы помимо слова в консоль выводился и его индекс.

3. Создать список A, который будет содержать размеры каждого слова в списке L1. Например, если первое слово списка L1 состоит из 10 символов, то первый элемент списка A будет иметь значение 10.
4. Определить размер четвертого слова в списке L1, используя при этом только список A.
5. Создать список B, который будет содержать только первые буквы каждого слова из списка L1. Например, если первое слово списка L1 «Монитор», то первый элемент списка B будет иметь значение «М».
6. Создать список C, как объединение пар элементов из двух списков A, B.
7. Объединить все элементы списка B в одну строку (один текст).

Задача № 19

1. Создать список L4 как копию списка L1. Удалить из списка L4 все слова, состоящие из 1 символа.
2. Удалить из списка L4 все слова, состоящие из 2 или 3 символов.
3. Создать список D, который будет содержать слова из списка L1, имеющие 5 или меньше символов.
4. Создать список E, который будет содержать слова из списка L1, имеющие 5 или больше символов, но не более 10 символов.
5. Создать список F, который будет содержать все остальные слова из списка L1, не вошедшие в список D или E.

Практическая работа № 3.

Циклы while, for в Python. Функции в Python

Часть № 1. Циклы while, for в Python

Задача № 1. Проход по элементам списка с помощью циклов for, while

Условие: имеется список L, состоящий из элементов 10, 15, 6, 13, 4, 2, 1, 2,

3.

Выполните следующие пункты задачи, предоставив ответ в двух вариантах:

1. Поочередно выведите все элементы списка L в консоль, используя:
 - a. цикл for
 - b. цикл while
2. Для каждого элемента списка прибавьте число 10 и выведите результат в консоль, используя:
 - a. цикл for
 - b. цикл while
3. Поочередно выводите все элементы списка L в консоль до тех пор, пока не встретится число 2, используя:
 - a. цикл for
 - b. цикл while

4. Поочередно выведите все элементы списка `L`, имеющие нечетные индексы в консоль, используя:
 - a. цикл `for`
 - b. цикл `while`

Задача № 2. Проход по элементам строки с помощью циклов `for`, `while`

Условие: имеется строка `s`, содержащая следующий текст: «Algorithmization and programming».

Выполните следующие пункты задачи, предоставив ответ в двух вариантах:

1. Поочередно выведите все элементы строки `s` в консоль, используя:
 - a. цикл `for`
 - b. цикл `while`
2. Поочередно выведите все буквы «а» строки `s` в консоль вместе с их индексами, используя:
 - a. цикл `for`
 - b. цикл `while`
3. Поочередно выведите все гласные буквы строки `s` в консоль вместе с их индексами, используя:
 - a. цикл `for`
 - b. цикл `while`

Задача № 3

Разработайте программу, которая в заданном списке имен находит самое длинное имя. Воспользуйтесь для этой цели циклом `for`.

Определите, как поведет себя алгоритм, предложенный вами в качестве решения, если «самых длинных» имен в списке будет несколько. В частности, как поведет себя ваш алгоритм, если все имена в списке будут одной длины?

Задача № 4

Модифицируйте программу из Задачи № 3 так, чтобы она находила самое длинное имя и самое короткое имя в заданном списке имен. Воспользуйтесь для этой цели так же циклом `for`.

Задача № 3

Разработайте программу, которая в заданном списке из пяти или более чисел находит 5 наименьших и 5 наибольших чисел.

Задача № 4

Используя только цикл `while` произведите подсчет элементов от значения `a` до значения `b` отсортированного списка целых чисел (произведите подсчет в диапазоне от `a` до `b`). При этом `a` и `b` могут не являться элементами списка.

Например, если имеется следующий список `L1` и значения `a = 2`, `b = 30`, программа должна дать ответ 5:

```
>>> L1 = [1, 2, 3, 10, 15, 20, 40, 100]
```

```
>>> a = 2
>>> b = 30
5
```

Задача № 5

Используя только цикл `for` решите Задачу № 4: произведите подсчет элементов от значения a до значения b отсортированного списка целых чисел (произведите подсчет в диапазоне от a до b). При этом a и b могут не являться элементами списка.

Часть № 2. Функции

Задача № 1. Основы функций

1. Создайте функцию суммирования двух чисел.
2. Создайте функцию для конкатенации двух строк.
3. Создайте функцию перемножения двух чисел.
4. Создайте функцию для возведения числа в степень, не используя при этом оператор `**`.
5. Создайте функцию, реализующую 7 любых арифметических операций. Для выбора операции предусмотреть специальный аргумент. Пример: если `arg1=='+'`, то выполняется сложение.
6. Создайте функцию, которая будет выводить в консоль текст n раз, где текст и n задается пользователем.
7. Создайте функцию для определения числа элементов в списке, не используя встроенную функцию `len()`.
8. Создайте функцию, которая на вход может получать произвольное количество аргументов и возвращает сумму переданных ей на вход чисел.

Пример:

```
In: func(2, 4, 1, 1)
```

```
Out: 8
```

```
In: func(4, 10)
```

```
Out: 14
```

9. Создайте глобальную переменную `PATH`, значение которой соответствует пути к некоторому каталогу «`C:\Downloads\Projects\Data`». Создайте функцию, которая на вход получает имя файла и изменяет значение глобальной переменной `PATH`, добавляя в конец пути обратную косую черту и имя файла.

Пример:

```
In: func('data1.txt')
```

```
Out: 'C:\Downloads\Projects\Data\data1.txt'
```

```
In: func('table.xlsx')
```

```
Out: 'C:\Downloads\Projects\Data\table.txt'
```

Задача № 2. Лямбда-функции

1. Создайте лямбда-функцию для суммирования двух элементов.
2. Создайте лямбда-функцию для получения возведения в квадрат переданного на вход числа.

3. Создайте лямбда-функцию, которая на вход получает 4 числа и возвращает результат их перемножения.
4. Создайте лямбда-функцию, которая на вход получает 1 число и возвращает результаты возведения этого числа в степень 2, степень 3 и степень 4.

Пример:

In: func(2)

Out: [4, 8, 16]

5. Создайте лямбда-функцию, которая на вход может получать произвольное число аргументов и возвращает число переданных ей на вход аргументов.

Пример:

In: func(1, 2, 10, 20)

Out: 4

Задача № 3

Возьмите текст из условия к задаче № 16 практического задания № 3.

1. Создайте функцию для очистки текста от знаков препинания.
2. Создайте функцию для подсчета числа слов в тексте без знаков препинания.
3. Создайте функцию для подсчета частоты встречаемости каждого слова, которая будет возвращать список слов и список частот. Например, если первое слово встречается 2 раза, то во втором списке первый элемент будет равен 2. Функция должна в консоли выводить указанные два списка в виде таблицы.
4. Создайте функцию для поиска слова в тексте.
5. Создайте функцию для поиска самого часто встречающегося слова в тексте. Функция должна возвращать слово и его частоту.

Задача № 4

Создайте функцию для упорядочивания слов в тексте по их частоте встречаемости. Функция должна возвращать список слов (от наиболее часто встречающегося к наименее часто встречающемуся или наоборот) и соответствующий список частот. Также функция должна в консоль выводить указанные два списка в виде таблицы.

Задача № 5

Разработайте функцию, которая в заданном списке имен находит самое длинное и самое короткое имя.

Задача № 6

Разработайте функцию, которая в заданном списке из пяти или более чисел находит 5 наименьших и 5 наибольших чисел.

Задача № 7

Создайте функцию для решения задачи «произвести подсчет элементов от значения a до значения b отсортированного списка целых чисел». При этом a и b могут не являться элементами списка. Реализуйте возможность выбора при вызове функции, какой цикл использовать (`while` или `for`).

Например, если имеется следующий список `L1` и значения $a = 2$, $b = 30$, то функция должна дать ответ 5:

```
>>> L1 = [1, 2, 3, 10, 15, 20, 40, 100]
>>> a = 2
>>> b = 30
>>> func(L1, a, b, 'for')
5
```

Задача № 8

Для функции из Задачи № 7 дополнительно реализуйте следующие возможности:

- a. корректная работа с неотсортированным списком (в этом случае список необходимо предварительно отсортировать);
- b. опциональный (по выбору пользователя) вывод в консоль подсчитываемых элементов списка и их индексов (чтобы пользователь мог выбрать, какую информацию выводить в консоль: ничего не выводить, выводить только элементы списка, выводить элементы списка вместе с их индексами);
- c. подсчет только четных элементов;
- d. помимо подсчета элементов в диапазоне от a до b , также реализовать вычисление:
 - суммы элементов;
 - произведения элементов;
 - арифметической средней элементов.

Задача № 9

Создайте функцию для работы пользователя с функцией из Задачи № 8 через консоль. Необходимо реализовать возможность ввода пользователем списка элементов, диапазон, дополнительные параметры (необходим ли подсчет только четных элементов и т.д.).

При вызове функции должен запускаться бесконечный цикл, в котором реализуется диалог с пользователем через ввод определенных команд в консоль. В таком случае выход из цикла будет производиться при вводе пользователем специальной команды, например слова `'end'`.

Практическая работа № 4. Классы в Python

Задача № 1

Условие: заказчику требуется разработать программное обеспечение (ПО) в виде некоторой образовательной платформы. Вам необходимо продумать, как может выглядеть модуль такого программного обеспечения, который отвечает за разграничение прав доступа к ресурсам ПО в зависимости от типа пользователя:

- User – стандартный тип пользователя.
- Student – тип пользователя с дополнительной информацией (о студенте).
- Teacher – тип пользователя с дополнительной информацией (о преподавателе) и с дополнительными возможностями.

Требуется:

1. Определите базовый класс User. В теле класса определите метод `show_info()`, который будет выводить в консоль информацию о пользователе. На данном этапе реализуйте вывод в консоль текста «Информация о пользователе:» при вызове `show_info()`.
2. Предусмотрите сохранение следующей информации о каждом новом пользователе:
 - ID пользователя формата «U001», который может использоваться как уникальный идентификатор пользователя.
 - Тип пользователя. Название типа пользователя соответствует имени класса, т.е. «User».
 - Логин.
 - Пароль.
3. Модифицируйте метод `show_info()` так, чтобы при его вызове в консоль выводилась вся информация о пользователе из п.2. (ID, Тип пользователя, Логин, Пароль).
4. Сделайте Логин и Пароль «приватными» атрибутами, доступ к которым предоставляется через соответствующие геттеры и сеттеры.
5. Определите производный класс Student (класс User будет являться базовым классом).
Предусмотрите сохранение следующей информации о каждом новом пользователе типа «Student»:
 - ID пользователя формата «S001», который может использоваться как уникальный идентификатор пользователя.
 - Тип пользователя, т.е. «Student».
 - Логин.
 - Пароль.
 - Имя.
 - Курс.
6. Определите производный класс Teacher (класс User будет являться базовым классом).
Предусмотрите сохранение следующей информации о каждом новом пользователе типа «Teacher»:
 - ID пользователя формата «T001», который может использоваться как уникальный идентификатор пользователя.
 - Тип пользователя, т.е. «Teacher».

- Логин.
- Пароль.
- Имя.
- Стаж работы.

Определите следующие методы класса `Teacher`:

- метод, с помощью которого пользователь типа «`Teacher`» может добавлять новых студентов в систему.
- метод, с помощью которого пользователь типа «`Teacher`» может удалять студентов из системы.

7. Реализуйте следующие дополнительные возможности:

- Реализуйте проверку на допустимый пароль (пароль должен содержать не менее 7 символов, включать цифры, прописные и заглавные буквы).
- Реализуйте проверку на допустимый логин (логин должен быть уникальным для каждого пользователя).
- Реализуйте автоматическое инкрементирование числовой части идентификатора пользователя (последние 3 цифры). Например, первому пользователю присваивается ID «U001», второму пользователю – «U002» и т.д.

При решении Задачи № 1 допускается в случае необходимости определять дополнительные классы, атрибуты и методы, использовать глобальные переменные и т.д.

Задача № 2

Определите класс `MyList` для описания встроенного списка Python (тип данных `list`) с некоторыми дополнительными возможностями:

- Определите метод `print_info()`, который выводит в консоль примерно следующую информацию о количестве элементов n в списке: «Данный список содержит n элемента(ов)».

Пример:

```
>>> list1 = MyList([1, 2, 3])
>>> list1.print_info()
'Данный список содержит 3 элемента(ов)'
```

- Определите метод `drop_value(value)`, который удаляет из списка все элементы, соответствующие `value`.

Пример:

```
>>> list1 = MyList([1, 2, 3, 2, 3, 40, 50, 2, 2, 1, 0, 2])
>>> list1.drop_value(2)
[1, 3, 3, 40, 50, 1, 0]
```

- Определите метод `add_indeces(value)`, который перед каждым элементом списка добавляет строковый элемент вида «`id: i`», где i –

индекс элемента в исходном списке. Реализуйте возможность использовать метод `append()` так, чтобы список сохранял подобный вид. Пример:

```
>>> list1 = MyList([1, 3, 9, 80])
>>> list1.add_indeces()
['id-0', 1, 'id-1', 3, 'id-2', 9, 'id-3', 80]
>>> list1.append(10)
>>> list1
['id-0', 1, 'id-1', 3, 'id-2', 9, 'id-3', 80, 'id-4', 10])
```

Практическая работа № 5.

Разработка и обучение моделей многослойного персептрона с использованием современных библиотек и фреймворков

Цель: изучить особенности разработки и обучения моделей многослойного персептрона с использованием современных библиотек и фреймворков.

Требуется:

1. Создайте матрицы с исходными данными, содержащими значения из исходного набора данных № 5. Подготовьте обучающие наборы данных и тестовые набор данных для решения задачи регрессии.
2. Используя библиотеки `scikit-learn`, `TensorFlow`, `PyTorch` создайте и обучите модели многослойного персептрона на обучающем наборе данных с разными значениями гиперпараметров, используя различные методы оптимизации, алгоритмы автоматизации подбора архитектур.
3. Протестируйте модели многослойного персептрона на тестовых наборах данных. Сделайте выводы о качестве моделей.

Практическая работа № 6.

Подбор оптимальных гиперпараметров для моделей многослойного персептрона при решении задачи регрессии

Продолжите разрабатывать программу из предыдущей работы.

1. Попробуйте улучшить результат работы нейронной сети, изменив ее конфигурацию, в частности:
 - a. Темп обучения.
 - b. Число эпох обучения.
 - c. Число нейронов в скрытом слое.
 - d.* *Диапазон значений весовых коэффициентов в соответствии с формулой (2).*
2. Найдите оптимальные значения параметров сети с помощью *следующих процедуры:*

- a. Поиск по сетке (Grid Search).
- b. Случайный поиск (Random Search).
- c. Кросс-валидация (Cross Validation).

Практическая работа № 7. Разработка рекуррентной нейронной сети (RNN)

Цель: изучить особенности разработки, обучения и применения рекуррентной нейронной сети (RNN) средствами Python.

Требуется:

1. Создайте матрицы с исходными данными, содержащими значения из исходного набора данных № 7. Подготовьте обучающие наборы данных и тестовые набор данных для решения задачи регрессии.
2. Используя библиотеки TensorFlow, PyTorch создайте и обучите модели рекуррентной нейронной сети (RNN) на обучающем наборе данных с разными значениями гиперпараметров, используя различные методы оптимизации, алгоритмы автоматизации подбора архитектур.
3. Протестируйте модели рекуррентной нейронной сети (RNN) на тестовых наборах данных. Сделайте выводы о качестве моделей.

Практическая работа № 8. Разработка сети долгой краткосрочной памяти (LSTM)

Цель: изучить особенности разработки, обучения и применения сети долгой краткосрочной памяти (LSTM) средствами Python.

Требуется:

1. Создайте матрицы с исходными данными, содержащими значения из исходного набора данных № 8. Подготовьте обучающие наборы данных и тестовые набор данных для решения задачи регрессии.
2. Используя библиотеки TensorFlow, PyTorch создайте и обучите модели сети долгой краткосрочной памяти (LSTM) на обучающем наборе данных с разными значениями гиперпараметров, используя различные методы оптимизации, алгоритмы автоматизации подбора архитектур.
3. Протестируйте модели сети долгой краткосрочной памяти (LSTM) на тестовых наборах данных. Сделайте выводы о качестве моделей.

6.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

Для оценки знаний, умений, навыков и формирования компетенции по дисциплине применяется **балльно-рейтинговая** система контроля и оценки успеваемости студентов.

В основу балльно-рейтинговой системы (БРС) положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущей работы в семестре.

В течение периода обучения по дисциплине студент должен выполнить и защитить 8 практических заданий (индивидуальных или групповых проектов), каждое из которых оценивается максимум на 10 баллов. За посещение занятий добавляется 0,33 балла за каждый час (максимум 10 баллов = $24 \cdot 0,42$), участие в конференции с докладом с использованием методов разработки элементов искусственного интеллекта – 10 баллов. Таким образом, максимально возможная сумма баллов равна: $8 \cdot 10 + 24 \cdot 0,42 + 10 + 10 = 80 + 10 + 10 = 100$.

До экзамена допускаются студенты, набравшие не менее 60% от максимального количества баллов, т.е. 60 баллов и более.

Предварительная оценка по дисциплине выставляется преподавателем в соответствии со шкалой:

Текущий рейтинг	Оценка			
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
в процентах	0-59	60-69	70-84	85-100
в баллах	0-59	60-69	70-84	85-100

Студенты, набравшие в течение семестра менее 60 баллов, пишут итоговую работу. К написанию итоговой работы допускаются студенты, **в случае выполнения всех практических работ.**

7. Учебно-методическое и информационное обеспечение дисциплины

7.1 Основная литература

1. Ростовцев, В. С. Искусственные нейронные сети : учебник для вузов / В. С. Ростовцев. — 5-е изд., стер. — Санкт-Петербург : Лань, 2025. — 216 с. — ISBN 978-5-507-50568-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/447392>. — Режим доступа: для авториз. пользователей.

2. Бессмертный, И. А. Системы искусственного интеллекта : учебник для вузов / И. А. Бессмертный. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 164 с. — (Высшее образование). — ISBN 978-5-534-18416-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561602>

3. Бессмертный, И. А. Интеллектуальные системы : учебник и практикум для вузов / И. А. Бессмертный, А. Б. Нугуманова, А. В. Платонов. — Москва : Издательство Юрайт, 2021. — 243 с. — (Высшее образование). — ISBN 978-5-534-01042-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/469867>

4. Воронов, М. В. Системы искусственного интеллекта : учебник и практикум для вузов / М. В. Воронов, В. И. Пименов, И. А. Небаев. — Москва : Издательство Юрайт, 2022. — 256 с. — (Высшее образование). — ISBN 978-5-534-14916-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/485440>

5. Иванов, В. М. Интеллектуальные системы : учебное пособие для среднего профессионального образования / В. М. Иванов ; под научной редакцией А. Н. Сесекина. — Москва : Издательство Юрайт, 2022. — 93 с. — (Профессиональное образование). — ISBN 978-5-534-07819-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/494505>

7.2 Дополнительная литература

1. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль ; перевод с английского А. А. Слинкина. — 2-е изд. — Москва : ДМК Пресс, 2018. — 652 с. — ISBN 978-5-97060-618-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/107901>. — Режим доступа: для авториз. пользователей.

2. Воронов, М. В. Системы искусственного интеллекта : учебник и практикум для вузов / М. В. Воронов, В. И. Пименов, И. А. Небаев. — Москва : Издательство Юрайт, 2022. — 256 с. — (Высшее образование). — ISBN 978-5-534-14916-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/485440>

3. Иванов, В. М. Интеллектуальные системы : учебное пособие для среднего профессионального образования / В. М. Иванов ; под научной

редакцией А. Н. Сесекина. — Москва : Издательство Юрайт, 2022. — 93 с. — (Профессиональное образование). — ISBN 978-5-534-07819-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/494505>

4. Станкевич, Л. А. Интеллектуальные системы и технологии : учебник и практикум для вузов / Л. А. Станкевич. — Москва : Издательство Юрайт, 2022. — 397 с. — (Высшее образование). — ISBN 978-5-534-02126-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489694>

5. Загорулько, Ю. А. Искусственный интеллект. Инженерия знаний: учебное пособие для вузов / Ю. А. Загорулько, Г.Б. Загорулько. - Москва: Издательство Юрайт, 2022. — 93 с. — (Высшее образование). — ISBN 978-5-534-07198-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/494205>

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

1. Machine Learning Crash Course. — URL: <https://developers.google.com/machine-learning/crash-course> (открытый доступ)

2. Цифровые профессии: Искусственный интеллект. — URL: <https://steps.2035.university/collections/f6361b9a-ea2e-41b1-a18f-9a2f84a9fcd4> (открытый доступ)

3. Kaggle. — URL: <https://www.kaggle.com/> (открытый доступ)

4. Machine Learning Repository. — URL: <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients> (открытый доступ)

5. TensorFlow library. <https://www.tensorflow.org/resources/libraries-extensions> (открытый доступ)

6. PyTorch. <https://pytorch.org/> (открытый доступ)

7. KERAS. <https://keras.io/> (открытый доступ)

8. dblp computer science bibliography: <https://dblp.uni-trier.de/db/about/index.html> (открытый доступ)

9. ICORE Conference Portal: https://portal.core.edu.au/conf-ranks/?search=A*+&by=all&source=CORE2023&sort=atitle&page=1 (открытый доступ)

9. Перечень программного обеспечения и информационных справочных систем

Таблица 9

Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины	Наименование программы	Тип программы	Автор	Год разработки
1	Разделы 1, 2	Python	расчетная, обучающая, контролирующая	Python Software Foundation	Текущая версия
2	Разделы 1, 2	Anaconda	расчетная, обучающая, контролирующая	Anaconda, Inc.	Текущая версия
3	Разделы 1, 2	Spyder	расчетная, обучающая, контролирующая	Spyder project contributors	Текущая версия
4	Разделы 1, 2	Microsoft Word	обучающая, контролирующая	Microsoft	Текущая версия
5	Разделы 1, 2	Microsoft Excel	обучающая	Microsoft	Текущая версия

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

Для проведения практических занятий нужен компьютерный класс с доступом в «Интернет», оснащенный программным обеспечением в соответствии с разделом 9.

Таблица 10

Сведения об обеспеченности специализированными аудиториями, кабинетами, лабораториями

Наименование специальных помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы
1	2
<i>учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (2й учебный корпус, 102 ауд.)</i>	<p>Количество рабочих мест: 16</p> <ol style="list-style-type: none"> Компьютеры 28 шт. Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11 a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE. Стенд «Сергеев Сергей Степанович 1910-1999» 1 шт. (Инв.№591013/25) Огнетушитель порошковый 1 шт. (Инв. №559527) Подвесное крепление к огнетушителю 1 шт. (Инв. № 559528) Жалюзи 2шт. (Инв. №1107-221225, Инв. №1107-221225) Стул 29 шт. Стол компьютерный 28 шт.

	<p>8. Стол для преподавателя 1 шт. 9. Доска маркерная 1 шт. 10. Трибуна напольная 1 шт. (без инв. №) Структурное подразделение: Институт Экономики и управления, Кафедра Статистики и кибернетики</p>
<p><i>учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (2й учебный корпус, 106 ауд.)</i></p>	<p>Количество рабочих мест: 16 Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE. Структурное подразделение: Институт Экономики и управления, Кафедра Статистики и кибернетики</p>
<p><i>учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации, помещение для самостоятельной работы (2й учебный корпус, 302 ауд.)</i></p>	<p>Количество рабочих мест: 16 1. Системный блок 17 шт. Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE. 2. Монитор 17 шт. 3. Телевизор 1 шт. 4. Стол для преподавателя 1 шт. 5. Стол компьютерный 16 шт. 6. Стул офисный 17 шт. Структурное подразделение: Институт Экономики и управления, Кафедра Статистики и кибернетики</p>
<p><i>учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (1й учебный корпус, 212 ауд.)</i></p>	<p>Количество рабочих мест: 24 Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE. Структурное подразделение: Кафедра Цифровая кафедра</p>
<p><i>учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения</i></p>	<p>Количество рабочих мест: 24 Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с)</p>

<i>курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (1й учебный корпус, 214 ауд.)</i>	и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE. Структурное подразделение: Кафедра Цифровая кафедра
<i>Центральная научная библиотека имени Н.И. Железнова</i>	Читальные залы библиотеки
<i>Студенческое общежитие</i>	Комната для самоподготовки

11. Методические рекомендации обучающимся по освоению дисциплины

Предполагается, что студент выполняет практическое задание в аудитории, дома оформляет и готовится по теоретическим вопросам к защите отчета на следующем занятии.

Виды и формы отработки пропущенных занятий

Студент, пропустивший занятие, обязан предъявить преподавателю документы установленного образца, подтверждающие необходимость пропуска. Не допускается пропуск занятий без уважительной причины.

Студент, пропустивший занятия, осваивает материал самостоятельно (выполняет практическое задание по своему варианту в компьютерном классе кафедры в часы, свободные от занятий, изучает теоретические вопросы).

Студент, пропустивший лекцию, отвечает на вопросы по пропущенной теме.

12. Методические рекомендации преподавателям по организации обучения по дисциплине

На первом занятии преподаватель закрепляет за каждым студентом номер варианта для выполнения индивидуальных работ (как правило, номер варианта соответствует порядковому номеру студента в журнале преподавателя). По каждой индивидуальной работе должна быть поставлена оценка по факту ее защиты. Защиту рекомендуется проводить на следующем после получения задания занятии. Преподаватель обязан проверить соответствие выполненного задания исходным данным варианта студента. Таким образом, исключается вероятность плагиата.

Преподаватель должен стимулировать студентов к занятию научно-исследовательской работой, изучению научной литературы по теме искусственного интеллекта, в т.ч. отечественной и зарубежной периодики.

Программу разработали:

Быков Д.В., старший преподаватель
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

Уколова А.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

РЕЦЕНЗИЯ

на рабочую программу дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении»
ОПОП ВО по направлению 05.04.04 «Гидрометеорология», направленность «Гидрометеорологическое обеспечение АПК»
(квалификация выпускника – магистр)

Гавриловской Надеждой Владимировной, канд. техн. наук, доцентом, и.о. заведующего кафедрой систем автоматизированного проектирования и инженерных расчетов (далее по тексту рецензент), проведена экспертиза рабочей программы дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении» ОПОП ВО по направлению 05.04.04 – «Гидрометеорология», направленность «Гидрометеорологическое обеспечение АПК», разработанной в ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», на кафедре статистики и кибернетики (разработчики – Быков Денис Витальевич, старший преподаватель кафедры статистики и кибернетики, Уколова Анна Владимировна, канд. экон. наук, доцент, и.о. заведующего кафедрой статистики и кибернетики).

Рассмотрев представленные на рецензию материалы, эксперт пришел к следующим выводам:

1. Предъявленная рабочая программа дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении» (далее по тексту Программа) соответствует требованиям ФГОС ВО по направлению 05.04.04 – «Гидрометеорология». Программа содержит все основные разделы, соответствует требованиям к нормативно-методическим документам.

2. Представленная в Программе **актуальность** учебной дисциплины в рамках реализации ОПОП ВО не подлежит сомнению – дисциплина относится к части учебного цикла, формируемой участниками образовательных отношений – Б1.В.

3. Представленные в Программе **цели** дисциплины соответствуют требованиям ФГОС ВО направления 05.04.04 – «Гидрометеорология».

4. В соответствии с Программой за дисциплиной «Искусственный интеллект в гидрометеорологическом обеспечении» закреплено 3 профессиональные **компетенции**, определяемые самостоятельно (**4 индикаторов**). Дисциплина «Искусственный интеллект в гидрометеорологическом обеспечении» и представленная Программа способна реализовать их в объявленных требованиях. Результаты обучения, представленные в Программе в категориях знать, уметь, владеть соответствуют специфике и содержанию дисциплины и демонстрируют возможность получения заявленных результатов.

5. Общая трудоёмкость дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении» составляет 3 зачётные единицы (108 часов).

6. Информация о взаимосвязи изучаемых дисциплин и вопросам исключения дублирования в содержании дисциплин соответствует действительности. Дисциплина «Искусственный интеллект в гидрометеорологическом обеспечении» взаимосвязана с другими дисциплинами ОПОП ВО и Учебного плана по направлению 05.04.04 – «Гидрометеорология» и возможность дублирования в содержании отсутствует.

7. Представленная Программа предполагает использование современных образовательных технологий, используемые при реализации различных видов учебной работы. Формы образовательных технологий соответствуют специфике дисциплины.

8. Программа дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении» предполагает проведение занятий в интерактивной форме.

9. Виды, содержание и трудоёмкость самостоятельной работы студентов, представленные в Программе, соответствуют требованиям к подготовке выпускников, содержащимся во ФГОС ВО направления 05.04.04 – «Гидрометеорология».

10. Представленные и описанные в Программе формы *текущей* оценки знаний (выполнение и защита практических заданий), соответствуют специфике дисциплины и требованиям к выпускникам.

Форма промежуточного контроля знаний студентов, предусмотренная Программой, осуществляется в форме зачета с оценкой, что соответствует статусу дисциплины, как дисциплины части учебного цикла, формируемой участниками образовательных отношений – Б1.В ФГОС ВО направления 05.04.04 – «Гидрометеорология».

11. Формы оценки знаний, представленные в Программе, соответствуют специфике дисциплины и требованиям к выпускникам.

12. Учебно-методическое обеспечение дисциплины представлено: основной литературой – 5 источников (базовый учебник), дополнительной литературой – 5 наименований, Интернет-ресурсы – 10 источников и соответствует требованиям ФГОС ВО направления 05.04.04 – «Гидрометеорология».

13. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении» и обеспечивает использование современных образовательных, в том числе интерактивных методов обучения.

14. Методические рекомендации студентам и методические рекомендации преподавателям по организации обучения по дисциплине дают представление о специфике обучения по дисциплине «Искусственный интеллект в гидрометеорологическом обеспечении».

ОБЩИЕ ВЫВОДЫ

На основании проведенной рецензии можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины «Искусственный интеллект в гидрометеорологическом обеспечении» ОПОП ВО по направлению 05.04.04 «Гидрометеорология», направленность «Гидрометеорологическое обеспечение АПК» (квалификация выпускника – магистр), разработанная Быковым Денисом Витальевичем, старшим преподавателем кафедры статистики и кибернетики, Уколовой Анной Владимировной, канд. экон. наук, доцентом, и.о. заведующего кафедрой статистики и кибернетики соответствует требованиям ФГОС ВО, современным требованиям экономики, рынка труда и позволит при её реализации успешно обеспечить формирование заявленных компетенций.

Рецензент:

Гавриловская Н.В., доцент кафедры систем автоматизированного проектирования и инженерных расчетов ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», кандидат технических наук

(ФИО, ученая степень, ученое звание)



(подпись)

«26» августа 2025 г.