

УТВЕРЖДАЮ  
Декан факультета

« \_\_\_\_ » \_\_\_\_\_ 20\_\_

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ

Направление/специальность подготовки	11.05.01 Радиоэлектронные системы и комплексы
Специализация/профиль/программа подготовки	Радиолокационные системы и комплексы
Уровень высшего образования	Специалитет
Форма обучения	Очная
Факультет	И Информационные и управляющие системы
Выпускающая кафедра	И4 РАДИОЭЛЕКТРОННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ
Кафедра-разработчик рабочей программы	Н2 Программная инженерия и интеллектуальные системы

КУРС	СЕМЕСТР	ОБЩАЯ ТРУДОЁМКОСТЬ (ЗАЧЕТНЫХ ЕДИНИЦ)	ЧАСЫ (по наличию видов занятий)									ВИД ПРОМЕЖУТОЧНОГО КОНТРОЛЯ
			ОБЩАЯ ТРУДОЁМКОСТЬ	АУДИТОРНЫЕ ЗАНЯТИЯ				САМОСТОЯТЕЛЬНАЯ РАБОТА				
				ВСЕГО	ЛЕКЦИИ	ЛАБОРАТОРНЫЙ ПРАКТИКУМ	ПРАКТИЧЕСКИЕ ЗАНЯТИЯ	ВСЕГО	КУРСОВОЙ ПРОЕКТ	КУРСОВАЯ РАБОТА	ДРУГИЕ ВИДЫ САМОСТ. РАБОТЫ	
1	2	4	144	68	34	0	34	76	0	0	76	ЭКЗ.

*ЛИСТ СОГЛАСОВАНИЯ*

**РАБОЧАЯ ПРОГРАММА СОСТАВЛЕНА В СООТВЕТСТВИИ С ТРЕБОВАНИЯМИ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО СТАНДАРТА ВЫСШЕГО ОБРАЗОВАНИЯ (ФГОС ВО)**

**11.05.01 Радиоэлектронные системы и комплексы**

год набора группы: 2026

Программу составил:

Кафедра Н2 Программная инженерия и интеллектуальные системы  
Лестенко Никита Александрович, старший преподаватель

\_\_\_\_\_

Программа рассмотрена  
на заседании кафедры-разработчика  
рабочей программы **Н2 Программная инженерия и интеллектуальные системы**

Заведующий кафедрой Семенова Е.Г., д.т.н., проф.

\_\_\_\_\_

Программа рассмотрена  
на заседании выпускающей кафедры

**И4 РАДИОЭЛЕКТРОННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ**

Заведующий кафедрой Сотникова Н.В., к.т.н., доц.

\_\_\_\_\_

# **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

## **ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ**

### **Разделы рабочей программы**

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО
3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ
4. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ
5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ
6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### **Приложения к рабочей программе дисциплины**

- Приложение 1. Аннотация рабочей программы
- Приложение 2. Технологии и формы обучения
- Приложение 3. Фонды оценочных средств

# 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является формирование следующих компетенций:

**ПК-93** — Способен генерировать новые идеи для решения задач цифровой экономики, абстрагироваться от стандартных моделей, перестраивать сложившиеся способы решения задач, выдвигать альтернативные варианты действий с целью выработки новых оптимальных алгоритмов

**ОПК-3** — Способен к логическому мышлению, обобщению, прогнозированию, постановке исследовательских задач и выбору путей их достижения, освоению работы на современном измерительном, диагностическом и технологическом оборудовании, используемом для решения различных научно-технических задач в области радиоэлектронной техники и информационно-коммуникационных технологий

**ОПК-7** — Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

**ОПК-8** — Способен использовать современные программные и инструментальные средства компьютерного моделирования для решения различных исследовательских и профессиональных задач

**ОПК-9** — Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения

Формированию компетенций служит достижение следующих результатов образования:

## **ПК-93**

*знания:*

знать алгоритмы и структуры данных для эффективного решения задач;;

*умения:*

уметь применять языки программирования для реализации программных решений;;

*навыки:*

обладать навыком отладки и оптимизации кода для повышения производительности и надежности программ;.

## **ОПК-3**

*знания:*

знать принципы логического мышления и обобщения применительно к постановке вычислительных задач;

*умения:*

уметь логически анализировать предметную задачу из области радиоэлектроники или ИКТ, выделять этапы её алгоритмизации и формулировать требования к интеллектуальной программной системе;

*навыки:*

обладать навыком написания программного кода для реализации интеллектуальных алгоритмов обработки, анализа и прогнозирования данных, включая использование ветвлений, циклов, функций, модулей и структур данных.

## **ОПК-7**

*знания:*

знать принципы организации современных информационных технологий, лежащих в основе анализа данных;

*умения:*

уметь использовать современные информационные технологии интеллектуального анализа данных для решения задач;

*навыки:*

обладать навыком разработки программ для построения эффективных интеллектуальных систем..

## **ОПК-8**

*знания:*

знать основы построения алгоритмов для решения задач.;

*умения:*

уметь применять алгоритмы и компьютерные программы для выполнения задач.;

*навыки:*

обладать навыком разрабатывать работающий программный код..

## **ОПК-9**

*знания:*

знать этапы жизненного цикла разработки программного обеспечения для интеллектуальных систем;

*умения:*

уметь разрабатывать алгоритмы для решения практических задач в области обработки данных и интеллектуального анализа;

*навыки:*

обладать навыком создания прикладных программных модулей, которые могут быть встроены в реальные информационно-коммуникационные системы.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина **ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ** является дисциплиной **обязательной части блока 1** программы подготовки по направлению *11.05.01 Радиоэлектронные системы и комплексы*.

Содержание дисциплины является логическим продолжением дисциплин: **ВВЕДЕНИЕ В КИБЕРФИЗИЧЕСКИЕ СИСТЕМЫ**.

Содержание дисциплины является основой для освоения дисциплин: **ИНТЕЛЛЕКТУАЛЬНЫЕ АВТОМАТИЗИРОВАННЫЕ ПРОИЗВОДСТВА, МЕТОДЫ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ В СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМАХ, МОДЕЛИРОВАНИЕ ПРОЦЕССОВ УПРАВЛЕНИЯ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫМИ И УПРАВЛЯЮЩИМИ СИСТЕМАМИ, ЦИФРОВАЯ ТРАНСФОРМАЦИЯ ПРОИЗВОДСТВЕННЫХ СИСТЕМ: МЕТОДЫ И ТЕХНОЛОГИИ, МЕТОДЫ ПРИНЯТИЯ ОПТИМИЗАЦИОННЫХ РЕШЕНИЙ**.

Предварительные компетенции, сформированные у обучающегося до начала изучения дисциплины:

- ОПК-3 — Способен к логическому мышлению, обобщению, прогнозированию, постановке исследовательских задач и выбору путей их достижения, освоению работы на современном измерительном, диагностическом и технологическом оборудовании, используемом для решения различных научно-технических задач в области радиоэлектронной техники и информационно-коммуникационных технологий
- ОПК-7 — Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности
- ОПК-8 — Способен использовать современные программные и инструментальные средства компьютерного моделирования для решения различных исследовательских и профессиональных задач
- ПК-94 — Способен к управлению информацией и данными, поиску источников информации и данных, восприятию, анализу, запоминанию и передаче информации с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач

### 3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 4 з.е., 144 ч.

#### 3.1. Содержание (дидактика) дисциплины

КУРС	СЕМЕСТР	Наименование разделов и дидактических единиц	ВСЕГО	Аудиторные занятия в контактной форме			Самостоятельная работа студентов	Формируемая компетенция, %				
				ВСЕГО	Лекции	Практические занятия		ПК-93	ОПК-3	ОПК-7	ОПК-8	ОПК-9
1	2	Раздел 1. Введение в Python. Основы синтаксиса, типы данных, ввод/вывод, линейные алгоритмы. Понятия информация и информационные технологии, программирование, язык и среда программирования. Этапы написания программы. Алгоритмы и блок-схемы. Общий обзор языка Python. Основы работы в IDLE. Базовый синтаксис и основные команды языка Python. Понятие переменной, требования к их наименованию.Целочисленные данные, вещественные данные, строковые данные, логические данные.	22	12	6	6	10	20	20	20	20	20
1	2	Раздел 2. Ветвления и циклы. Условные операторы, циклы for, while, рекуррентные вычисления, оператор break.	34	16	8	8	18	20	20	20	20	20
1	2	Раздел 3. Структуры данных. Списки, кортежи, словари, множества. Базовые алгоритмы обработки.Функции и методы работы со списками, кортежами, словарями, множествами. Срезы.	16	8	4	4	8	20	20	20	20	20
1	2	Раздел 4. Функции и модули. Определение функций, аргументы, область видимости, lambda-функции, импорт библиотек.	38	20	10	10	18	20	20	20	20	20
1	2	Раздел 5. Библиотеки для интеллектуального анализа данных. Изучение библиотек math, pandas, numpy,mathprolib, random. Работа с данными с импользованием библиотек math, pandas, numpy,mathprolib, random.	34	12	6	6	22	20	20	20	20	20
Всего за 2 семестр			144	68	34	34	76	100	100	100	100	100
Всего по дисциплине			144	68	34	34	76	100	100	100	100	100

#### 3.2. Аудиторный практикум

№ п/п	Номер и наименование раздела дисциплины	Тема практического занятия	Объем, ауд. часов
1	Раздел 1. Введение в Python.	Установка среды (VS Code, PyCharm, IDLE Python, Jupiter). Первая программа: ввод/вывод, типы данных, вычисления.	6
2	Раздел 2. Ветвления и циклы.	Решение задач на ветвления и циклы( числовые ряды, табуляция функций)	8
3	Раздел 3. Структуры данных.	Обработка списков: поиск, сортировка, срезы. Работа со словарями.	4
4	Раздел 4. Функции и модули.	Создание пользовательских функций. Передача параметров. Функции высшего порядка.	10
5	Раздел 5. Библиотеки для интеллектуального анализа данных.	Визуализация данных (mathprolib). Вычисления с numpy и pandas. Генерация случайных данных.	6
<b>Всего за 2 семестр</b>			<b>34</b>

#### 3.3. Самостоятельная работа студента (СРС)

№ п/п	Номер и наименование раздела дисциплины	Содержание учебного задания	Объем, часов
1	Раздел 1. Введение в Python.	Изучение предусмотренной программой дисциплины рекомендуемой литературы	4
2		Оформление отчета	2
3		Подготовка к практическим занятиям	4
4	Раздел 2. Ветвления и циклы.	Изучение предусмотренной программой	4

		дисциплины рекомендуемой литературы	
5		Подготовка к практическим занятиям	12
6		Оформление отчета	2
7	Раздел 3. Структуры данных.	Подготовка к практическим занятиям	4
8		Оформление отчета	2
9		Изучение предусмотренной программой дисциплины рекомендуемой литературы	2
10	Раздел 4. Функции и модули.	Подготовка к практическим занятиям	14
11		Оформление отчета	2
12		Изучение предусмотренной программой дисциплины рекомендуемой литературы	2
13	Раздел 5. Библиотеки для интеллектуального анализа данных.	Подготовка к практическим занятиям	18
14		Оформление отчета	2
15		Изучение предусмотренной программой дисциплины рекомендуемой литературы	2
Всего за 2 семестр			76

#### 4. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

СЕМЕСТР	НЕДЕЛИ СЕМЕСТРА																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2			ИПЗ			ДР	ИПЗ		ИПЗ	ДР			ИПЗ			ДР	ИПЗ

Условные обозначения:

- ДР – диагностическая работа;
- ИПЗ – индивидуальное практическое задание.

**Текущий контроль успеваемости** студентов проводится в дискретные временные интервалы в следующих формах:

- диагностическая работа;
- индивидуальное практическое задание.

**Промежуточная аттестация** проводится в формах:

- экзамен.



## 5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 5.1. Основная литература по дисциплине:

1. Д. Бейдер. . Чистый Python. Тонкости программирования для профи. Санкт-Петербург: Питер, 2021, эл. рес.
2. Д. Ю. Фёдоров. . Программирование на языке высокого уровня Python. Москва: Юрайт, 2023, эл. рес.
3. Н. А. Лестенко, И. Д. Мамаев. Технология обработки данных для предметно-ориентированных задач на языке Python. БГТУ "ВОЕНМЕХ" им. Д.Ф. Устинова, 2024, 12 экз.
4. Н. А. Лестенко, И. Д. Мамаев. Технология обработки данных для предметно-ориентированных задач на языке Python. БГТУ "ВОЕНМЕХ" им. Д.Ф. Устинова, 2024, эл. рес.
5. Ф. Шолле. . Глубокое обучение на Python. Санкт-Петербург: Питер, 2021, эл. рес.

### 5.2. Дополнительная литература по дисциплине:

не требуется.

### 5.3. Периодические издания:

1. Прикладная информатика.

### 5.4. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины, электронные библиотечные системы:

1. <https://e.lanbook.com/> — ЭБС Лань;;
2. <https://urait.ru/> — Образовательная платформа «Юрайт». Для вузов и ссузов.;

### Современные профессиональные базы данных:

1. <https://rusneb.ru> – Национальная электронная библиотека (НЭБ);
2. <https://cyberleninka.ru/> - Научная электронная библиотека «Киберленинка»;  
<http://www.rfbr.ru/rffi/ru/library> - Полнотекстовая электронная библиотека Российского фонда фундаментальных исследований.

### Информационные справочные системы:

1. Техэксперт – Информационный портал технического регулирования: Нормы, правила, стандарты РФ;
2. [http://library.voenmeh.ru/jirbis2/index.php?option=com\\_irbis&view=irbis&Itemid=457](http://library.voenmeh.ru/jirbis2/index.php?option=com_irbis&view=irbis&Itemid=457) - БД ГОСТов собственной генерации БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова;
3. <http://www.consultant.ru/>- КонсультантПлюс- информационный портал правовой информации.

### 5.5. Программное обеспечение:

1. Набор средств трансляции, компоновки, отладки и выполнения Python 3.x с интегрированной средой разработки IDLE.

### 5.6. Информационные технологии:

взаимодействие с обучающимися посредством ЭИОС Moodle БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова.

## **6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **6.1. Лекционные занятия:**

специализированные требования по оборудованию отсутствуют; аудитория с посадочными местами по количеству студентов; доска.

### **6.2. Практические занятия:**

1. Проектор;
2. Набор средств трансляции, компоновки, отладки и выполнения Python 3.x с интегрированной средой разработки IDLE.

### **6.3. Прочее:**

1. рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;
2. рабочие места студентов, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде.

### **Аннотация рабочей программы**

Дисциплина **ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ** является дисциплиной **обязательной части блока 1** программы подготовки по направлению *11.05.01 Радиоэлектронные системы и комплексы*. Дисциплина реализуется на факультете *Н Робототехника и инновационная инженерия* БГТУ "ВОЕНМЕХ" им. Д.Ф. Устинова кафедрой *Н2 Программная инженерия и интеллектуальные системы*.

Дисциплина нацелена на формирование *компетенций*:

ПК-93 Способен генерировать новые идеи для решения задач цифровой экономики, абстрагироваться от стандартных моделей, перестраивать сложившиеся способы решения задач, выдвигать альтернативные варианты действий с целью выработки новых оптимальных алгоритмов;

ОПК-3 Способен к логическому мышлению, обобщению, прогнозированию, постановке исследовательских задач и выбору путей их достижения, освоению работы на современном измерительном, диагностическом и технологическом оборудовании, используемом для решения различных научно-технических задач в области радиоэлектронной техники и информационно-коммуникационных технологий;

ОПК-7 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности;

ОПК-8 Способен использовать современные программные и инструментальные средства компьютерного моделирования для решения различных исследовательских и профессиональных задач;

ОПК-9 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

Содержание дисциплины охватывает круг вопросов, связанных с разработкой алгоритмов, созданием программного обеспечения и автоматизацией процессов для решения практических задач с помощью вычислительных систем.

Программой дисциплины предусмотрены следующие **виды контроля**:

**Текущий контроль успеваемости** студентов проводится в дискретные временные интервалы в следующих формах:

- диагностическая работа;
- индивидуальное практическое задание.

**Промежуточная аттестация** проводится в формах:

- экзамен.

Общая трудоемкость освоения дисциплины составляет **4 з.е., 144 ч**. Программой дисциплины предусмотрены лекционные занятия (**34 ч.**), практические занятия (**34 ч.**), самостоятельная работа студента (**76 ч.**).

## ТЕХНОЛОГИИ И ФОРМЫ ОБУЧЕНИЯ

### Рекомендации по освоению дисциплины для студента

Трудоемкость освоения дисциплины составляет 144 ч., из них 68 ч. аудиторных занятий, и 76 ч., отведенных на самостоятельную работу студента.

Рекомендации по распределению учебного времени по видам самостоятельной работы и разделам дисциплины приведены в таблице.

Контроль освоения дисциплины производится в соответствии с Положением о текущем, рубежном контроле успеваемости и промежуточной аттестации обучающихся.

Формы контроля и критерии оценивания приведены в приложении 3 к Рабочей программе.

Наименование работы	Рекомендуемая литература	Трудоемкость, час.
Раздел 1. Введение в Python.		
Изучение предусмотренной программой дисциплины рекомендуемой литературы	Н. А. Лестенко, И. Д. Мамаев. Технология обработки данных для предметно-ориентированных задач на языке Python: БГТУ "ВОЕНМЕХ" им. Д.Ф. Устинова, 2024 (1-6)	4
Оформление отчета		2
Подготовка к практическим занятиям		4
Итого по разделу 1		10
Раздел 2. Ветвления и циклы.		
Изучение предусмотренной программой дисциплины рекомендуемой литературы	Ф. Шолле. . Глубокое обучение на Python: Санкт-Петербург: Питер, 2021 (1-2)	4
Подготовка к практическим занятиям		12
Оформление отчета		2
Итого по разделу 2		18
Раздел 3. Структуры данных.		
Подготовка к практическим занятиям	Д. Бейдер. . Чистый Python. Тонкости программирования для профи: Санкт-Петербург: Питер, 2021 (2-4)	4
Оформление отчета		2
Изучение предусмотренной программой дисциплины рекомендуемой литературы		2
Итого по разделу 3		8
Раздел 4. Функции и модули.		
Подготовка к практическим занятиям	Д. Ю. Фёдоров. . Программирование на языке высокого уровня Python: Москва: Юрайт, 2023 (3-4)	14
Оформление отчета		2
Изучение предусмотренной программой дисциплины рекомендуемой литературы		2
Итого по разделу 4		18
Раздел 5. Библиотеки для интеллектуального анализа данных.		
Подготовка к практическим занятиям	Н. А. Лестенко, И. Д. Мамаев. Технология обработки данных для предметно-ориентированных задач на языке Python: БГТУ "ВОЕНМЕХ" им. Д.Ф. Устинова, 2024 (4)	18
Оформление отчета		2
Изучение предусмотренной программой дисциплины		2

рекомендуемой литературы		
Итого по разделу 5		22

## **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

Фонд оценочных средств, позволяющие оценить результаты обучения по данной дисциплине, включают в себя:

- диагностическая работа
- индивидуальное практическое задание;
- экзамен.

### **Критерии оценивания**

#### **Диагностическая работа**

Диагностическая работа проводится в форме теста в ЭИОС Moodle:

- при правильном ответе менее чем на 60% вопросов - не аттестация;
- при правильном ответе на 60% вопросов и более - аттестация.

#### **Индивидуальное практическое задание**

Перечень индивидуальных заданий:

1. Введение в Python.
2. Ветвления и циклы.
3. Структуры данных.
4. Функции и модули.
5. Библиотеки для интеллектуального анализа данных.

Индивидуальное задание считается выполненным и защищенным успешно при условии:

- наличия программного кода, реализующего поставленную задачу;
- наличия отчета;
- защиты индивидуального задания по комплекту тестовых вопросов для защиты индивидуальных заданий, размещенного в УМК дисциплины.

#### **Экзамен**

График контрольных мероприятий предусматривает выполнение студентом пяти индивидуальных заданий, каждое из которых может быть оценено согласно технологической карте дисциплины. Также в рамках курса проводятся три диагностические работы, каждая из которых оценивается на 10 баллов при условии успешного прохождения.

Оценка за экзамен выставляется по сумме результатов контрольных мероприятий, проводимых в течение семестра. Максимальная сумма баллов за семестр – 100 баллов.

Набранная итоговая сумма баллов пересчитывается в оценку по следующей схеме:

- 60 – 74 баллов – удовлетворительно;
- 75 – 84 балла - хорошо;
- 85– 100 баллов – отлично.

В случае несогласия студента с оценкой, выставляемой согласно БРС, может быть проведён экзамен, вопросы к которому располагаются в УМК дисциплины. В этом случае экзамен проходит по билетам. Каждый билет содержит два вопроса и задачу. Корректный ответ на один вопрос и частично решенную задачу - "удовлетворительно", Полные ответы на два вопроса и частично решенная задача - "хорошо", либо неполные ответы на два вопроса и полностью корректно решенная задача - "хорошо" . Развернутый ответ на два вопроса и полностью корректно решенная задача- "отлично"

Паспорт фонда оценочных средств

КУРС	СЕМЕСТР	Наименование разделов и дидактических единиц	ВСЕГО	Аудиторные занятия в контактной форме			Самостоятельная работа студентов	Формируемая компетенция, %					НАИМЕНОВАНИЕ ОЦЕНОЧНОГО СРЕДСТВА
				ВСЕГО	Лекции	Практические занятия		ПК-93	ОПК-3	ОПК-7	ОПК-8	ОПК-9	
1	2	Раздел 1. Введение в Python.	22	12	6	6	10	20	20	20	20	20	Индивидуальное практическое задание
1	2	Раздел 2. Ветвления и циклы.	34	16	8	8	18	20	20	20	20	20	Индивидуальное практическое задание
1	2	Раздел 3. Структуры данных.	16	8	4	4	8	20	20	20	20	20	Индивидуальное практическое задание
1	2	Раздел 4. Функции и модули.	38	20	10	10	18	20	20	20	20	20	Индивидуальное практическое задание
1	2	Раздел 5. Библиотеки для интеллектуального анализа данных.	34	12	6	6	22	20	20	20	20	20	Индивидуальное практическое задание
Всего за 2 семестр			144	68	34	34	76	100	100	100	100	100	
Всего по дисциплине			144	68	34	34	76	100	100	100	100	100	

## Оценочные материалы по дисциплине ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ

**ПК-93 - Способен генерировать новые идеи для решения задач цифровой экономики, абстрагироваться от стандартных моделей, перестраивать сложившиеся способы решения задач, выдвигать альтернативные варианты действий с целью выработки новых оптимальных алгоритмов**

№ 1 Прочитайте текст и запишите развернутый обоснованный ответ

Чем отличается кортеж (tuple) от списка (list) при передаче в функцию, которая должна гарантировать неизменность переданных данных для последующего интеллектуального анализа?

№ 2 Прочитайте текст и запишите развернутый обоснованный ответ

Допустим, вы используете библиотеку `sklearn.pipeline.Pipeline` для построения конвейера обработки данных. В одном из шагов кастомного трансформера вы обращаетесь к глобальной переменной, которая изменяется в другом модуле во время выполнения `fit()`. При повторном вызове `predict()` с теми же данными результаты оказываются разными, хотя трансформер не содержит случайных компонент. Какая фундаментальная особенность Python (связанная с изменяемыми объектами и областью видимости) приводит к такому эффекту, и как переписать трансформер, чтобы обеспечить детерминированность?

№ 3 Прочитайте текст и установите соответствие

**Сопоставьте фрагмент кода слева с его результатом или поведением справа (при условии, что `import pandas as pd` выполнен).**

### Левый столбец

1. `df[df['x'] > 0]`

2.

`df['x'].apply(lambda v: v if v>0 else 0)`

3. `for i in`

`range(len(df)):`

`if df.loc[i, 'x'] < 0:`

`df.loc[i, 'x'] = 0`

### Правый столбец

А. Замена всех значений меньше 0 на 0 с помощью цикла

Б. Фильтрация строк, где значение в столбце 'x' положительное

В. Векторизованная замена отрицательных значений на 0

Г. Поэлементное преобразование с помощью лямбда-функции

№ 4 Прочитайте текст и установите соответствие

**Сопоставьте понятие или механизм слева с его описанием справа.**

### Левый столбец

1. Область видимости (scope)

2. Замыкание (closure)

3. `globals()`

### Правый столбец

А. Механизм, позволяющий функции «запоминать» переменные из внешней функции даже после её завершения

Б. Словарь, который хранит все глобальные переменные модуля

В. Правила, определяющие, где переменная доступна (локальная, объемлющая, глобальная, встроенная)

Г. Источник трудноуловимых ошибок, так как объект сохраняет своё состояние между вызовами функции

№ 5 Прочитайте текст и установите последовательность

**Дан фрагмент кода с ошибкой в порядке операций. Расположите строки в правильном порядке, чтобы программа вычисляла факториал числа 5 рекурсивно и корректно обрабатывала случай нуля:**

**Перемешанные строки:**

А. `else:`



B. def factorial(n):  
C. return 1  
D. if n == 0:  
E. print(factorial(5))  
F. return n \* factorial(n-1)

№ 6 Прочитайте текст и установите последовательность

**Выполняется следующий код. Расположите промежуточные состояния DataFrame в том порядке, в котором они возникают в процессе выполнения цепочки операций:**

```
import pandas as pd

df = pd.DataFrame({'A': [1, 2, None, 4], 'B': [5, None, 7, 8]})

df = df.dropna().assign(C=lambda x: x['A'] + x['B']).query('C > 10').reset_index(drop=True)
```

**Варианты состояний (условные описания):**

DataFrame с колонками A, B и новой колонкой C, содержащей сумму A+B, только для строк без пропусков

Исходный DataFrame с пропусками (None) в A и B

DataFrame только с теми строками, где C > 10, сброшенным индексом

DataFrame после удаления всех строк, содержащих хотя бы один None

№ 7 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

**Дан список nums = [1, 2, 3, 4, 5]. Каким будет результат выполнения [x for x in nums if x % 2 == 0 else 0]?**

Варианты ответа:

[0, 2, 0, 4, 0]

[2, 4]

[0, 0, 0, 0, 0]

Синтаксическая ошибка

№ 8 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

**Что произойдёт при выполнении кода?**

```
def outer():

    x = 10

    def inner():

        print(x)

        x = 20

    inner()

outer()
```

Варианты ответа:

Выведет 10

Выведет 20

Выведет 10 и затем 20

Ошибка UnboundLocalError

- № 9 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

**Нужно объединить два DataFrame: df1 (сотрудники) и df2 (отделы) по ключу 'dept\_id'. В df1 есть строки с dept\_id, отсутствующими в df2, и вы хотите сохранить ВСЕ строки из df1, даже если нет соответствия в df2. Какой параметр how в pd.merge() нужно использовать?**

Варианты ответа:

how='inner'

how='outer'

how='left'

how='right'

- № 10 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

**Какие из следующих выражений корректно создают список [0, 2, 4, 6, 8] в Python?**

Варианты ответа:

list(range(0, 10, 2))

[i for i in range(10) if i % 2 == 0]

[0, 2, 4, 6, 8]

list(filter(lambda x: x % 2 == 0, range(10)))

- № 11 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

**Какие из следующих утверждений об области видимости (scope) в Python являются верными?**

Варианты ответа:

Переменная, определённая внутри функции, доступна глобально после её вызова

Ключевое слово global позволяет изменять глобальную переменную внутри функции

Переменные, определённые в цикле for, доступны после цикла на том же уровне вложенности

nonlocal используется для изменения переменной из объемлющей (не глобальной) области видимости

- № 12 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

**Какие из следующих утверждений о сравнении pandas.Series и numpy.ndarray являются верными?**

Варианты ответа:

Оба поддерживают векторные операции (+, -, \*, /)

Series может иметь индекс произвольного типа (не только целочисленный)

ndarray автоматически выравнивает данные по индексу при арифметических операциях

В Series можно хранить разнородные типы данных в одном объекте

**ОПК-3 - Способен к логическому мышлению, обобщению, прогнозированию, постановке исследовательских задач и выбору путей их достижения, освоению работы на современном измерительном, диагностическом и технологическом оборудовании, используемом для решения**

№ 1 Прочитайте текст и запишите развернутый обоснованный ответ

Напишите **псевдокод** или опишите логику декоратора `@retry(max_attempts=3, delay=1)`, который автоматически повторяет выполнение функции при возникновении исключения. Декоратор должен принимать параметры (количество попыток и задержку между попытками), логировать каждую попытку (выводить номер попытки и тип ошибки), а после исчерпания всех попыток — выбрасывать последнее исключение. Объясните, почему внешняя функция декоратора (`retry`) возвращает внутреннюю функцию-декоратор, а не сразу обёртку.

№ 2 Прочитайте текст и запишите развернутый обоснованный ответ

При работе с большим DataFrame (миллионы строк) вы заметили, что операция `df[df['status'] == 'active']` выполняется быстро, а последующая группировка `groupby('region')` на отфильтрованных данных — медленно. Почему фильтрация не ускоряет группировку так, как ожидалось? Как, используя категориальный тип (`dtype='category'`) для столбца `region`, можно улучшить производительность? Ответ дайте в двух-трёх предложениях.

№ 3 Прочитайте текст и установите соответствие

**Сопоставьте оператор или выражение (левый столбец) с его временной сложностью в худшем случае (правый столбец) для стандартных типов CPython.**

**Левый столбец**

1. `x in list`
2. `x in set`
3. `del dict[key]`
4. `list.insert(0, val)`

**Правый столбец**

- A.  $O(1)$
- Б.  $O(n)$
- В.  $O(\log n)$
- Г.  $O(n^2)$

№ 4 Прочитайте текст и установите соответствие

**Сопоставьте фрагмент кода (левый столбец) с его результатом или поведением (правый столбец). Предполагается `import numpy as np`.**

**Левый столбец**

1. `np.array([1,2,3]) * 2`
2. `np.array([1,2,3]) + np.array([4,5])`
3. `np.array([1,2,3]).repeat(2)`
4. `np.arange(5)[::-1]`

**Правый столбец**

- A. `[2 4 6]` (массив)
- Б. `[1 2 3 1 2 3]`
- В. Ошибка: несовпадение размеров
- Г. `[4 3 2 1 0]`

№ 5 Прочитайте текст и установите последовательность

**Расположите строки кода в правильном порядке, чтобы программа вычисляла сумму чисел от 1 до N (N вводится пользователем) с использованием цикла `while`, корректно обрабатывала `N < 1` (выводила 0) и содержала обработку ошибки ввода (если введено не число).**

**Перемешанные строки:**

- A. `except ValueError:`
- Б. `n = int(input("Введите N: "))`
- В. `total = 0`
- Г. `print("Ошибка: введите целое число")`
- Д. `i = 1`
- Е. `while i <= n:`
- Ж. `i += 1`

H. total += i  
I. print(f"Сумма от 1 до {n} = {total}")  
J. try:  
K. if n < 1: total = 0; break (или аналогично, но в цикл до входа)

№ 6 Прочитайте текст и установите последовательность

**Расположите этапы вычисления рекурсивной функции factorial(5) в порядке вызовов и возвратов. Функция определена так:**

python

```
def factorial(n): if n <= 1: return 1 return n * factorial(n-1)
```

**Перемешанные события:**

A. factorial(5) вызывает factorial(4)  
B. factorial(2) вызывает factorial(1)  
C. factorial(1) возвращает 1  
D. factorial(3) вызывает factorial(2)  
E. factorial(4) вызывает factorial(3)  
F. factorial(2) возвращает 2  
G. factorial(3) возвращает 6  
H. factorial(5) возвращает 120  
I. factorial(4) возвращает 24

№ 7 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Что выведет код?

python

```
x = 5 if x > 5: print("A") elif x == 5: print("B") else: print("C")
```

**Варианты:** 1) A, 2) B, 3) C, 4) ничего

№ 8 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какой метод DataFrame вернёт количество строк и столбцов в виде кортежа?

**Варианты:**

df.shape

df.size

df.ndim

df.count()

№ 9 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какая из следующих операций изменит исходный список a = [1, 2, 3]?

**Варианты:**

b = a + [4]

a.append(4)

a = [1, 2, 3, 4]

print(a[0])

№ 10 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из следующих вариантов являются корректными способами создания списка от 0 до 4 (включительно)? Выберите все.

**Варианты:**`list(range(5))``[0, 1, 2, 3, 4]``[x for x in range(5)]``(0, 1, 2, 3, 4)`

№ 11 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из утверждений об аргументах функций в Python верны? (Выберите все)

**Варианты:**

Аргументы по умолчанию могут быть изменяемыми объектами

Именованные аргументы всегда должны следовать после позиционных

\*args собирает лишние именованные аргументы

Функция может возвращать несколько значений через запятую

№ 12 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из следующих операций создадут массив NumPy нулевой длины? Выберите все.

**Варианты:**`np.array([])``np.empty(0)``np.zeros(0)``np.arange(0)`

**ОПК-7 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности**

№ 1 Прочитайте текст и запишите развернутый обоснованный ответ

Разработайте (словесно) декоратор `@cache_result(ttl=60)`, который кэширует результат функции на определённое время (`ttl` — время жизни в секундах). Кэш должен храниться в словаре, где ключ — кортеж аргументов, значение — кортеж (результат, время\_истечения). При повторном вызове проверяется, не истекло ли время; если истекло — результат пересчитывается. Объясните, как обеспечить потокобезопасность (без блокировок) для однопото

№ 2 Прочитайте текст и запишите развернутый обоснованный ответ

**Вопрос:**

Вы импортировали большой набор данных через `pd.read_csv('data.csv')`. Обнаружилось, что столбцы имеют неправильные типы: числовые столбцы загрузились как `object` из-за наличия символов ? вместо пропусков. Как, используя параметры `na_values` и `dtype` в `read_csv`, а также последующее преобразование, корректно загрузить данные за один проход (без повторного чтения файла)? Объясните кратко.

№ 3 Прочитайте текст и установите соответствие

**Сопоставьте фрагмент кода (левый столбец) с его результатом (правый столбец).**

**Левый столбец**

1. `[x**2 for x in range(3)]`    А. `[0, 1, 4]`

2. `[x for x in range(5) if x % 2 == 0]`    Б. `[0, 2, 4]`

**Правый столбец**

**Левый столбец****Правый столбец**

3.  
`list(map(lambda B: [2, 3, 4]  
 x: x+1, [1,2,3]))`  
 4.  
`list(filter(lambda Г: [3, 4]  
 x: x>2, [1,2,3,4]))`

№ 4 Прочитайте текст и установите соответствие

**Сопоставьте метод pandas со его действием (предполагается стандартный DataFrame df).**

**Левый столбец****Правый столбец**

1. <code>df.drop_duplicates()</code>	A. Удаляет строки с пропущенными значениями
2. <code>df.dropna()</code>	Б. Удаляет дубликаты строк
3. <code>df.fillna(0)</code>	В. Заменяет пропуски на 0
4. <code>df.rename(columns=</code>	Г. Переименовывает столбец 'a' в 'b'
<code>{'a':'b'})</code>	

№ 5 Прочитайте текст и установите последовательность

**Расположите строки кода в правильном порядке, чтобы получить функцию `fib(n)`, возвращающую n-е число Фибоначчи ( $n \geq 0$ ) с использованием рекурсии и мемоизации (через `functools.lru_cache`).**

**Перемешанные строки:**

A. `return fib(n-1) + fib(n-2)`  
 B. `from functools import lru_cache`  
 C. `def fib(n):`  
 D. `if n < 2: return n`  
 E. `@lru_cache(maxsize=None)`

№ 6 Прочитайте текст и установите последовательность

**Расположите этапы выполнения следующего кода в хронологическом порядке (от первого действия к последнему) с учётом возникновения исключения:**

python

```
try: a = int("abc") b = 10 / a except ValueError: print("ValueError") except ZeroDivisionError:
print("ZeroDivision") else: print("No error") finally: print("Finally")
```

**Перемешанные события:**

Вычисление выражения `int("abc")`

Выброс исключения `ValueError`

Переход в блок `except ValueError`

Печать `"ValueError"`

Пропуск блока `else`

Выполнение блока `finally`

Печать `"Finally"`

Завершение программы

№ 7 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какое значение будет у переменной x после выполнения?

```
python
```

```
x = 3 x += 5 x *= 2
```

**Варианты:** 1) 8, 2) 16, 3) 13, 4) 10

№ 8 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Сколько раз выполнится тело цикла?

```
python
```

```
for i in range(3, 0, -1): print(i)
```

**Варианты:** 1) 3, 2) 2, 3) 4, 4) бесконечно

№ 9 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какой из следующих вариантов создаст массив NumPy из пяти нулей?

**Варианты:**

1. np.zeros(5)
2. np.empty(5)
3. np.ones(5)
4. np.array(5)

№ 10 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из следующих структур данных в Python являются **изменяемыми (mutable)**? Выберите все.

**Варианты:**

1. list
2. tuple
3. dict
4. set
5. str

№ 11 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из следующих определений функций в Python синтаксически корректны? (Выберите все)

**Варианты:**

1. def f(): pass
2. def f(x, y=[]): pass
3. def f(\*args, \*\*kwargs): pass
4. def f(x, \*y, z): pass

№ 12 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из этих операций можно применить к объекту Series в pandas? (Выберите все)

**Варианты:**

1. series.mean()
2. series.sort\_values()
3. series.plot() (если импортирован matplotlib)
4. series.resize()

**ОПК-8 - Способен использовать современные программные и инструментальные средства компьютерного моделирования для решения различных исследовательских и профессиональных задач**

№ 1 Прочитайте текст и запишите развернутый обоснованный ответ

Почему в анализе данных часто используют связку функций `map()` или `apply()` с лямбда-выражениями вместо явного цикла `for` по строкам `DataFrame` из библиотеки `pandas`?

№ 2 Прочитайте текст и запишите развернутый обоснованный ответ

В вашем проекте по интеллектуальному анализу данных есть модуль, который кэширует результаты тяжёлых вычислений в глобальном словаре. Однако при параллельной обработке (например, с помощью `concurrent.futures.ThreadPoolExecutor`) разные потоки одновременно читают и пишут в этот словарь, что приводит к состоянию гонки и нестабильным результатам. Как, не отказываясь от глобального кэша и не используя блокировки на каждой операции чтения, можно перестроить структуру данных и функции доступа, чтобы обеспечить потокобезопасность и при этом сохранить высокую производительность?

№ 3 Прочитайте текст и установите соответствие

**Сопоставьте библиотеку или её компонент слева с её основным назначением справа.**

**Левый столбец**

**Правый столбец**

- |                                   |   |
|-----------------------------------|---|
| 1. <code>pandas.DataFrame</code>  | А. Линейная алгебра и работа с многомерными массивами                     |
| 2. <code>numpy.ndarray</code>     | Б. Классификация, регрессия, кластеризация (алгоритмы машинного обучения) |
| 3. <code>matplotlib.pyplot</code> | В. Табличные данные с индексацией строк и столбцов, обработка пропусков   |
|                                   | Г. Построение графиков, гистограмм, scatter-диаграмм                      |

№ 4 Прочитайте текст и установите соответствие

**Сопоставьте операцию или функцию слева с её временной сложностью (в среднем) для соответствующей структуры данных справа.**

**Левый столбец**

**Правый столбец**

- |   |                         |
|---|-------------------------|
| 1. Поиск элемента в <code>list</code> по значению ( <code>x in lst</code> ) | А. $O(1)$               |
| 2. Вставка элемента в <code>dict</code> ( <code>d[key] = value</code> )     | Б. $O(n)$               |
| 3. Проверка наличия элемента в <code>set</code> ( <code>elem in s</code> )  | В. $O(1)$ (хеш-таблица) |
|   | Г. $O(1)$               |

№ 5 Прочитайте текст и установите последовательность

**Расположите строки кода в правильном порядке, чтобы получить функцию, которая возвращает список чётных чисел из переданного списка, используя генератор списка:**

**Варианты (строки кода):**

- A. `return [x for x in numbers if x % 2 == 0]`
- B. `def get_even(numbers):`
- C. `result = get_even([1, 2, 3, 4, 5])`
- D. `print(result)`

№ 6 Прочитайте текст и установите последовательность



**Вам нужно построить конвейер (Pipeline) в scikit-learn для следующих этапов обработки и обучения. Расположите действия в том порядке, который обеспечит корректную работу без утечки данных из тестовой выборки в обучение:**

**Варианты последовательности:**

Применение StandardScaler к тестовой выборке (используя параметры, обученные на обучающей выборке)

Разделение данных на обучающую и тестовую выборки через train\_test\_split

Применение StandardScaler.fit\_transform() к обучающей выборке

Импорт Pipeline из sklearn.pipeline

Обучение модели (например, LogisticRegression) на масштабированных обучающих данных

Создание пайплайна: Pipeline([('scaler', StandardScaler()), ('clf', LogisticRegression())])

Предсказание на тестовой выборке с помощью обученного пайплайна

Вызов метода .fit() на пайплайне с передачей только обучающих данных

№ 7 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

**Что выведет следующий код?**

```
def func(x, lst=[]):
```

```
    lst.append(x)
```

```
    return lst
```

```
print(func(1))
```

```
print(func(2))
```

Варианты ответа:

[1] и [2]

[1] и [1, 2]

[1] и [1]

Ошибка: изменяемый аргумент по умолчанию запрещён

№ 8 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

**Каким образом наиболее эффективно отфильтровать строки DataFrame df по условию df['age'] > 30 и одновременно выбрать только столбцы ['name', 'city']?**

Варианты ответа:

```
df[df['age'] > 30][['name', 'city']]
```

```
df.loc[df['age'] > 30, ['name', 'city']]
```

```
df.query('age > 30')[['name', 'city']]
```

```
df.filter(like='name').filter(like='city')
```

№ 9 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

**Какая из следующих операций выполняется «на месте» (in-place) и не создаёт новый массив NumPy?**

Варианты ответа:

`a = a + 1`

`a += 1`

`np.append(a, 1)`

`a = a * 2`

№ 10 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

**Какие из следующих утверждений о функциях в Python являются верными?**

Варианты ответа:

Функция может возвращать несколько значений, упакованных в кортеж

Аргументы по умолчанию вычисляются при каждом вызове функции

\*args позволяет передавать произвольное количество именованных аргументов

\*\*kwargs собирает необъявленные именованные аргументы в словарь

№ 11 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Вопрос 5 (Тема: Библиотеки для интеллектуального анализа данных — pandas)

**Какие из следующих операций с pandas DataFrame создают новый объект (возвращают копию), а не изменяют исходный DataFrame?**

Варианты ответа:

`df.dropna(inplace=True)`

`df.dropna()`

`df.fillna(0, inplace=False)`

`df['new'] = df['old'] * 2`

№ 12 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Вопрос 6 (Тема: Библиотеки для интеллектуального анализа данных — общее)

**Какие библиотеки из стандартного набора для интеллектуального анализа данных в Python предоставляют средства для визуализации данных?**

Варианты ответа:

`matplotlib.pyplot`

`seaborn`

`numpy`

`plotly.express`

**ОПК-9 - Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения**

№ 1 Прочитайте текст и запишите развернутый обоснованный ответ

Вы работаете с DataFrame, содержащим столбцы price и quantity. Вам нужно добавить новый столбец total как произведение price \* quantity, но только для строк, где quantity > 0. Если quantity

$\leq 0$ , то total должен быть равен NaN. Как наиболее эффективно и читаемо сделать это в pandas, избегая явного цикла? Приведите пример кода и поясните, почему такой подход быстрее.

№ 2 Прочитайте текст и запишите развернутый обоснованный ответ

Представьте, что у вас есть большой текстовый файл (несколько ГБ), каждая строка которого содержит целое число. Напишите **описание генератора** `read_integers(file_path)`, который читает файл построчно, преобразует каждую строку в целое число и возвращает его, а также пропускает строки, которые не являются целыми числами (логируя ошибку в отдельный файл). Объясните, почему генератор лучше, чем чтение всего файла в список, с точки зрения потребления памяти.

№ 3 Прочитайте текст и установите соответствие

**Сопоставьте выражение (левый столбец) с его результатом (правый столбец).**

**Левый столбец**

**Правый столбец**

- |                               |                        |
|-------------------------------|------------------------|
| 1. <code>3 * 'a' + 'b'</code> | A. <code>'aaab'</code> |
| 2. <code>'a' in 'abc'</code>  | B. <code>True</code>   |
| 3. <code>'abc'[1:3]</code>    | B. <code>'bc'</code>   |
| 4. <code>len('hello')</code>  | Г. <code>5</code>      |

№ 4 Прочитайте текст и установите соответствие

**Сопоставьте имя модели из scikit-learn (левый столбец) с типовой задачей (правый столбец).**

**Левый столбец**

**Правый столбец**

- |  |                                   |
|--|-----------------------------------|
| 1. <code>LinearRegression</code>       | A. Классификация                  |
| 2. <code>LogisticRegression</code>     | Б. Регрессия (предсказание числа) |
| 3. <code>KMeans</code>                 | В. Кластеризация                  |
| 4. <code>RandomForestClassifier</code> | Г. Классификация (ансамбль)       |

№ 5 Прочитайте текст и установите последовательность

**Расположите строки кода в правильном порядке, чтобы программа находила максимальное значение в списке numbers без использования встроенной функции `max()` (алгоритм линейного поиска).**

**Перемешанные строки:**

- A. `for num in numbers:`
- B. `numbers = [3, 1, 4, 1, 5, 9, 2]`
- C. `max_val = numbers[0]`
- D. `print(f"Максимум: {max_val}")`
- E. `if num > max_val:`
- F. `max_val = num`

№ 6 Прочитайте текст и установите последовательность

**Расположите этапы выполнения следующего кода в порядке обращения к переменным (поиск по LEGB-правилу).**

`x = 'global'`

`def outer():`

`x = 'outer'`

`def inner():`

`x = 'inner'`

`print(x)`

inner()

outer()

**Перемешанные события (обращения к x):**

1. Обращение к x внутри print в inner()
2. Поиск в локальной области inner — находит 'inner'
3. Вывод 'inner'
4. После завершения inner — возврат в outer
5. Создание локальной переменной x в outer со значением 'outer'
6. Создание локальной переменной x в inner со значением 'inner'
7. Определение глобального x = 'global'

№ 7 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

**Вопрос:**

Что будет выведено?

```
a = [1, 2]
```

```
b = a
```

```
a = a + [3]
```

```
print(b)
```

**Варианты:**

[1, 2, 3]

[1, 2]

[1, 2, 3, 3]

Ошибка

№ 8 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какое значение примет s после цикла?

```
s = 0
```

```
for i in range(1, 4):
```

```
    s += i
```

**Варианты:** 1) 3, 2) 4, 3) 6, 4) 10

№ 9 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какая функция из matplotlib.pyplot создаёт круговую диаграмму?

**Варианты:**

1. plot()
2. pie()
3. bar()
4. hist()

№ 10 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из следующих операций над множеством set в Python являются допустимыми? (Выберите все.)

**Варианты:**

1. Добавление элемента (add)
2. Удаление элемента (remove)
3. Доступ по индексу (s[0])
4. Проверка на вхождение (elem in s)

№ 11 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из следующих вызовов функции `def func(a, b=2, *args): pass` являются корректными? (Выберите все.)

**Варианты:**

1. `func(1)`
2. `func(1, 3)`
3. `func(1, b=4, 5)`
4. `func(1, 2, 3, 4)`

№ 12 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из следующих вызовов функции `def func(a, b=2, *args): pass` являются корректными? (Выберите все.)

**Варианты:**

1. `func(1)`
2. `func(1, 3)`
3. `func(1, b=4, 5)`
4. `func(1, 2, 3, 4)`