

УТВЕРЖДАЮ
 Декан факультета

«___» _____ 20__

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ

Направление/специальность подготовки	17.05.02 Стрелково-пушечное, артиллерийское и ракетное оружие
Специализация/профиль/программа подготовки	Роботизированные комплексы вооружения
Уровень высшего образования	Специалитет
Форма обучения	Очная
Факультет	Е Оружие и системы вооружения
Выпускающая кафедра	Е1 СТРЕЛКОВО-ПУШЕЧНОЕ, АРТИЛЛЕРИЙСКОЕ И РАКЕТНОЕ ОРУЖИЕ
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)**

17.05.02 Стрелково-пушечное, артиллерийское и ракетное оружие

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

Е1 СТРЕЛКОВО-ПУШЕЧНОЕ, АРТИЛЛЕРИЙСКОЕ И РАКЕТНОЕ ОРУЖИЕ

Заведующий кафедрой Алешин А.С., к.т.н.

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:

ОПК-6 — Способен использовать в инженерной деятельности методы, способы и средства получения, хранения, переработки информации с использованием современных информационных технологий

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

Формированию компетенций служит достижение следующих результатов образования:

ОПК-6

знания:

знать основы построения алгоритмов для эффективного решения задач;

умения:

уметь применять языки программирования для решения конкретных задач;

навыки:

обладать навыком поиска и обработки информации для решения требуемой задачи.

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;.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина **ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ** является дисциплиной **обязательной части блока 1** программы подготовки по направлению *17.05.02 Стрелково-пушечное, артиллерийское и ракетное оружие*.

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, МЕТОДЫ ПРИНЯТИЯ ОПТИМИЗАЦИОННЫХ РЕШЕНИЙ, ВЫПОЛНЕНИЕ, ПОДГОТОВКА К ПРОЦЕДУРЕ ЗАЩИТЫ И ЗАЩИТА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ, МОДЕЛИРОВАНИЕ ПРОЦЕССОВ УПРАВЛЕНИЯ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫМИ И УПРАВЛЯЮЩИМИ СИСТЕМАМИ, ЦИФРОВАЯ ТРАНСФОРМАЦИЯ ПРОИЗВОДСТВЕННЫХ СИСТЕМ: МЕТОДЫ И ТЕХНОЛОГИИ**.

Предварительные компетенции, сформированные у обучающегося до начала изучения дисциплины:

- ОПК-3 — Способен понимать сущность и значение информации в развитии современного информационного общества, осознавать опасность и угрозы, возникающие в процессе этого развития, соблюдать основные требования информационной безопасности, в том числе защиты государственной тайны
- 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		ОПК-6	ПК-93
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
1	2	Раздел 2. Ветвления и циклы. Условные операторы, циклы for, while, рекуррентные вычисления, оператор break.	34	16	8	8	18	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
1	2	Раздел 4. Функции и модули. Определение функций, аргументы, область видимости, lambda-функции, импорт библиотек.	38	20	10	10	18	20	20
1	2	Раздел 5. Библиотеки для интеллектуального анализа данных. Изучение библиотек math, pandas, numpy, mathprolib, random. Работа с данными с использованием библиотек math, pandas, numpy, mathprolib, random.	34	12	6	6	22	20	20
Всего за 2 семестр			144	68	34	34	76	100	100
Всего по дисциплине			144	68	34	34	76	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
Всего за 2 семестр			34

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 дисциплины рекомендуемой литературы	4
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://urait.ru/> — Образовательная платформа «Юрайт». Для вузов и ссузов.;;
2. <https://e.lanbook.com/> — ЭБС Лань;.

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 - 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 *17.05.02 Стрелково-пушечное, артиллерийское и ракетное оружие*. Дисциплина реализуется на факультете *Н Робототехника и инновационная инженерия* БГТУ "ВОЕНМЕХ" им. Д.Ф. Устинова кафедрами *Н2 Программная инженерия и интеллектуальные системы*.

Дисциплина нацелена на формирование *компетенций*:

ОПК-6 Способен использовать в инженерной деятельности методы, способы и средства получения, хранения, переработки информации с использованием современных информационных технологий;

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.

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		ОПК-6	ПК-93	
1	2	Раздел 1. Введение в Python.	22	12	6	6	10	20	20	Индивидуальное практическое задание
1	2	Раздел 2. Ветвления и циклы.	34	16	8	8	18	20	20	Индивидуальное практическое задание
1	2	Раздел 3. Структуры данных.	16	8	4	4	8	20	20	Индивидуальное практическое задание
1	2	Раздел 4. Функции и модули.	38	20	10	10	18	20	20	Индивидуальное практическое задание
1	2	Раздел 5. Библиотеки для интеллектуального анализа данных.	34	12	6	6	22	20	20	Индивидуальное практическое задание
Всего за 2 семестр			144	68	34	34	76	100	100	
Всего по дисциплине			144	68	34	34	76	100	100	

Оценочные материалы по дисциплине ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ

ОПК-6 - Способен использовать в инженерной деятельности методы, способы и средства получения, хранения, переработки информации с использованием современных информационных технологий

№ 1 Прочитайте текст и запишите развернутый обоснованный ответ
Чем отличается использование оператора break от continue внутри цикла while или for, и в каком сценарии каждый из них предпочтительнее?

№ 2 Прочитайте текст и установите последовательность

Расположите следующие этапы в правильном порядке для реализации кастомного трансформера в scikit-learn, который должен:

наследоваться от BaseEstimator и TransformerMixin

в fit() вычислять и сохранять среднее и стандартное отклонение для каждого признака

в transform() использовать сохранённые параметры для стандартизации данных

корректно работать в Pipeline (без утечки данных)

Перемешанные этапы:

A. Определение метода transform(self, X), который применяет сохранённые mean_ и std_ к переданным данным: $(X - \text{self.mean_}) / \text{self.std_}$

B. Импорт BaseEstimator и TransformerMixin из sklearn.base

C. Определение метода fit(self, X, y=None), который вычисляет $\text{self.mean_} = X.\text{mean}(\text{axis}=0)$ и $\text{self.std_} = X.\text{std}(\text{axis}=0)$, а затем возвращает self

D. Создание класса class StandardizeTransformer(BaseEstimator, TransformerMixin):

E. Добавление проверки в transform, чтобы избежать деления на ноль (например, замена нулевых std_ на 1 или выброс предупреждения)

F. Возврат преобразованных данных из transform

G. Определение метода __init__(self) (даже пустого) для соответствия API scikit-learn

№ 3 Прочитайте текст и установите последовательность

Расположите этапы обработки данных в правильной последовательности, которая минимизирует использование памяти и избегает SettingWithCopyWarning в pandas. Задача: загрузить большой CSV-файл (10 ГБ), отфильтровать строки, где age > 18, создать новый столбец age_group на основе age (категории: 'young' для 19–35, 'middle' для 36–55, 'senior' для 56+), затем агрегировать среднюю зарплату по группам и сохранить результат в новый CSV.

Перемешанные этапы:

A. `df['age_group'] = pd.cut(df['age'], bins=[18, 35, 55, 150], labels=['young', 'middle', 'senior'])`

B. `df = df[df['age'] > 18]` (фильтрация)

C. `result = df.groupby('age_group')['salary'].mean().reset_index()`

D. `df = pd.read_csv('large_file.csv', usecols=['age', 'salary'])` (загрузка только нужных колонок для экономии памяти)

E. `result.to_csv('result.csv', index=False)`

F. `del df` (явное удаление промежуточного DataFrame для освобождения памяти)

G. df = df.сорu() после фильтрации для предотвращения цепочечного индексирования

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

Каким будет результат выполнения следующего кода?

```
python
```

```
a = 10 b = a a = 20 print(b)
```

Варианты ответов:

20

10

None

Ошибка: переменная b не определена

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

Вопрос:

Сколько раз выполнится тело цикла в следующем фрагменте?

```
python
```

```
count = 0 for i in range(3): for j in range(2): count += 1
```

Варианты ответов:

3

5

6

2

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

Вопрос:

Какой метод pandas используется для просмотра первых 5 строк DataFrame df?

Варианты ответов:

```
df.tail()
```

```
df.head()
```

```
df.info()
```

```
df.describe()
```

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

Тема: Введение в Python

Вопрос:

Какие из следующих типов данных в Python являются **неизменяемыми (immutable)**? (Выберите все верные варианты.)

Варианты ответов:

list

tuple

dict

str

int

set

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

Какие из следующих операций **корректно** удаляют элемент 'x' из списка `my_list = ['a', 'b', 'x', 'c', 'x']`? (Выберите **все верные варианты**.)

Варианты ответов:

`my_list.remove('x')`

`del my_list[2]`

`my_list.pop(2)`

`my_list.delete('x')`

`my_list.remove_all('x')`

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

Вопрос:

Какие из следующих методов pandas **возвращают новый DataFrame** (не изменяют исходный), если не указан параметр `inplace=True`? (Выберите **все верные варианты**.)

Варианты ответов:

`df.dropna()`

`df.fillna(0)`

`df.reset_index()`

`df.sort_values(by='col')`

`df.drop_duplicates()`

№ 10 Прочитайте текст и запишите развернутый обоснованный ответ

Дан массив NumPy `data` формы (1000000, 50) и вектор меток `labels` длины 1000000 с тремя возможными значениями (-1, 0, 1). Требуется заменить все значения в `data` по правилу: если метка строки равна -1, то все элементы строки заменить на их абсолютное значение; если метка равна 0 — заменить на 0; если метка равна 1 — оставить без изменений. Запрещены явные циклы на Python. Как это сделать эффективно, используя только векторизованные операции NumPy и, возможно, `np.where` или индексацию? Почему предложенное вами решение будет работать быстрее наивного цикла?

№ 11 Прочитайте текст и установите соответствие

Сопоставьте каждую конструкцию или подход из левого столбца с её наиболее точным описанием в правом столбце. В каждом описании упоминаются особенности работы с памятью, областью видимости или производительностью.

Левый столбец

1. Генератор списка (`list comprehension`) внутри функции, которая вызывается многократно

Правый столбец

А. Создаёт новый объект каждый раз при вызове; может привести к высокому потреблению памяти при больших объёмах данных, но работает быстро благодаря внутренней оптимизации CPython

Левый столбец

Правый столбец

2. Функция-генератор с <code>yield</code> , обрабатывающая строки <code>DataFrame</code>	Б. Позволяет организовать ленивые вычисления и обрабатывать данные порциями, не загружая весь результат в память; идеально для потоковой обработки больших объёмов
3. Кэширование результатов с помощью <code>functools.lru_cache</code> для функции, аргументом которой является изменяемый список	В. Вызовет ошибку типа (<code>TypeError</code>), так как список не является хешируемым; для кэширования необходимо преобразовать список в кортеж или использовать неизменяемый тип
4. Использование <code>global</code> словаря для накопления промежуточных статистик внутри цикла обработки	Г. Может привести к неявным побочным эффектам и трудностям отладки, особенно при параллельной обработке; предпочтительнее передавать словарь как аргумент или использовать замыкание

№ 12 Прочитайте текст и установите соответствие

Сопоставьте фрагмент кода (слева) с его скрытой проблемой или особенностью производительности (справа). Все фрагменты предназначены для фильтрации и преобразования данных в `pandas/NumPy`.

Левый столбец (фрагмент кода)

Правый столбец (проблема / особенность)

1. <code>for i in range(len(df)): if df.iloc[i, 'value'] > 0: df.iloc[i, 'value'] = df.iloc[i, 'value'] ** 2</code>	А. Цепочечное индексирование, которое может привести к предупреждению и непредсказуемому поведению (работа с копией вместо оригинала)
2. <code>df[df['value'] > 0] ['value'] = df[df['value'] > 0]['value'] ** 2</code>	Б. Избыточное копирование данных при каждой итерации из-за повторного вызова <code>df.iloc</code> ; работает крайне медленно на больших <code>DataFrame</code>
3. <code>df.loc[df['value'] > 0, 'value'] = df.loc[df['value'] > 0, 'value'] ** 2</code>	В. Нет проблем — это правильный векторизованный способ изменения столбца по условию с использованием одного прохода <code>loc</code>
4. <code>df['value'].where(df['value'] > 0, other=0, inplace=True)</code>	Г. Изменяет данные, но не возводит в квадрат положительные значения, а заменяет отрицательные на 0 — поведение не соответствует остальным фрагментам

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

Правый столбец

- | | |
|---|---|
| 1. <code>df[df['x'] > 0]</code> | A. Замена всех значений меньше 0 на 0 с помощью цикла |
| 2. <code>df['x'].apply(lambda v: v if v>0 else 0)</code> | Б. Фильтрация строк, где значение в столбце 'x' положительное |
| 3. <code>for i in range(len(df)): if df.loc[i, 'x'] < 0: df.loc[i, 'x'] = 0</code> | 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

Правый столбец

- | | |
|------------------------------|--|
| 1. Область видимости (scope) | A. Механизм, позволяющий функции «запоминать» переменные из внешней функции даже после её завершения |
| 2. Замыкание (closure) | Б. Словарь, который хранит все глобальные переменные модуля |
| 3. <code>globals()</code> | 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:

- A. `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, 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

№ 7 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

№ 8 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

№ 9 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

№ 10 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'

№ 11 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)))

№ 12 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