

УТВЕРЖДАЮ  
Декан факультета

« \_\_\_\_ » \_\_\_\_\_ 20\_\_

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ

Направление/специальность подготовки	17.05.01 Боеприпасы и взрыватели
Специализация/профиль/программа подготовки	Патроны и гильзы
Уровень высшего образования	Специалитет
Форма обучения	Очная
Факультет	Е Оружие и системы вооружения
Выпускающая кафедра	Е4 Технология патронного производства и обработка металлов давлением
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)**

**17.05.01 Боеприпасы и взрыватели**

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

**Е4 Технология патронного производства и обработка металлов давлением**

Заведующий кафедрой Нестеров Н.И., 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:

**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-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-3**

*знания:*

знать основные принципы функционирования информационных систем и роль программного обеспечения в информационном обществе;

*умения:*

уметь оценивать потенциальные риски и угрозы безопасности при разработке и использовании программных продуктов;

*навыки:*

применять требования информационной безопасности при тестировании и отладке программ.

### **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 *17.05.01 Боеприпасы и взрыватели*.

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**.

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-3	ПК-93
1	2	<b>Раздел 1. Введение в Python.</b> 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	<b>Раздел 2. Ветвления и циклы.</b> Условные операторы, циклы for, while, рекуррентные вычисления, оператор break.	34	16	8	8	18	20	20
1	2	<b>Раздел 3. Структуры данных.</b> 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	<b>Раздел 4. Функции и модули.</b> Определение функций, аргументы, область видимости, lambda-функции, импорт библиотек.	38	20	10	10	18	20	20
1	2	<b>Раздел 5. Библиотеки для интеллектуального анализа данных.</b> 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
<b>Всего за 2 семестр</b>			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](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 *17.05.01 Боеприпасы и взрыватели*. 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*:

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-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-3	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-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**

№ 1 Прочитайте текст и запишите развернутый обоснованный ответ

есть функция `process_data(data, method='mean')`, которая должна либо возвращать среднее (`method='mean'`), либо медиану (`method='median'`) каждого столбца переданного DataFrame. Однако вы замечаете, что при повторных вызовах с одним и тем же DataFrame и разными `method` функция каждый раз заново вычисляет агрегацию, что неэффективно. Как, используя механизм кэширования (например, `functools.lru_cache` или собственный словарь), можно модифицировать функцию так, чтобы результаты для каждого метода сохранялись, но при этом функция корректно работала с изменяемым типом DataFrame? Опишите решение в двух-трёх предложениях.

№ 2 Прочитайте текст и запишите развернутый обоснованный ответ

Дан очень большой лог-файл (сотни гигабайт) с записями вида `timestamp, user_id, action`. Требуется написать эффективный по памяти генератор, который:

Читает файл построчно (без загрузки всего файла в память).

Фильтрует строки, оставляя только действия пользователя с заданным `user_id`.

Группирует записи по часам (по полю `timestamp`), возвращая для каждого часа количество действий этого пользователя.

При этом генератор не должен хранить все часы в памяти — выдавать результат по мере накопления данных за очередной час.

Опишите структуру генератора и логику его работы. Почему использование обычного списка для накопления записей перед группировкой было бы неэффективным?

№ 3 Прочитайте текст и установите соответствие

**Сопоставьте метод или атрибут `pandas` (левый столбец) с его наиболее точным описанием и временной сложностью в типичном случае (правый столбец).**

### Левый столбец

1. `df.loc[row_label, col_label]`

2. `df.iloc[row_index, col_index]`

3. `df[['col1', 'col2']]`

4. `df.dropna()`

### Правый столбец

А.  $O(n)$  по памяти — создаёт копию указанных столбцов; используется для уменьшения размера данных

Б.  $O(1)$  по времени, доступ по меткам строки и столбца с использованием внутренних индексных массивов

В.  $O(1)$  по времени, доступ по целочисленной позиции, аналогично списку Python

Г.  $O(n)$  по времени и памяти — удаление строк с пропусками требует полного прохода и создания нового DataFrame

№ 4 Прочитайте текст и установите соответствие

**Сопоставьте конструкцию или механизм (левый столбец) с примером кода, где эта конструкция используется или проявляется наиболее характерным образом (правый столбец).**

### Левый столбец

### Правый столбец

**Левый столбец****Правый столбец**

- |                           |   |
|---------------------------|---|
| 1. Замыкание (closure)    | A. <code>def outer(x): def inner(y): return x + y; return inner; add5 = outer(5); print(add5(3))</code> |
| 2. Декоратор              | Б. <code>@timer def long_running(): time.sleep(1); print('done')</code>                                 |
| 3. Лямбда-выражение       | В. <code>sorted(data, key=lambda x: x['age'])</code>  |
| 4. Генераторное выражение | Г. <code>sum(x**2 for x in range(1000000) if x % 2 == 0)</code>   |

№ 5 Прочитайте текст и установите последовательность

Расположите этапы создания **пользовательского трансформера для scikit-learn**, который наследуется от `BaseEstimator` и `TransformerMixin`

в `fit()` вычисляет минимум и максимум каждого признака

в `transform()` выполняет нормализацию в диапазон `[0, 1]` по формуле  $(X - \min) / (\max - \min)$

корректно обрабатывает случай, когда `min == max` (возвращает нулевой столбец)

**Перемешанные этапы:**

- A. Импорт `BaseEstimator`, `TransformerMixin` из `sklearn.base`
- B. Определение метода `fit(self, X, y=None)`, вычисление `self.min_ = X.min(axis=0)` и `self.max_ = X.max(axis=0)`, возврат `self`
- C. Создание класса `class MinMaxScalerCustom(BaseEstimator, TransformerMixin):`
- D. Определение метода `transform(self, X)`, вычисление `range_ = self.max_ - self.min_`, замена нулевых `range_` на 1 (чтобы избежать деления на ноль), возврат  $(X - self.min_) / range_$
- E. Определение метода `__init__(self)` (пустого) для совместимости с `scikit-learn`
- F. Возврат `self` из `transform` (хотя обычно возвращаются данные, здесь важно: `transform` возвращает преобразованный массив)

№ 6 Прочитайте текст и установите последовательность

Расположите шаги в правильном порядке для реализации **итеративного (без рекурсии) обхода бинарного дерева в порядке «in-order» (лево-корень-право)** с использованием явного стека. Каждый шаг описывает **логическое действие** или состояние алгоритма.

**Перемешанные шаги:**

Если стек пуст и `current == None` — завершить алгоритм

`current = current.right`

Вывести значение `current.val` (обработать узел)

Инициализировать пустой стек и `current = root`

`current = current.left`

Положить `current` в стек и установить `current = current.left`

Извлечь узел из стека в `current`

Пока `current != None`: выполнить шаг 6

Перейти к шагу 3, если стек не пуст

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

**Вопрос:**

Какое значение будет выведено на экран после выполнения кода?

python

`x = [1, 2, 3] y = x x = [4, 5, 6] print(y[0])`



**Варианты:** 1) 4, 2) 1, 3) [1,2,3], 4) Ошибка

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

Что выведет этот код?

python

```
for i in range(3): if i == 1: continue print(i)
```

**Варианты:** 1) 0 2, 2) 0 1 2, 3) 1, 4) 0

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

Какая функция из NumPy создаёт массив из равномерно распределённых значений в заданном интервале с **указанным количеством элементов**?

**Варианты:** 1) np.arange(), 2) np.linspace(), 3) np.random.rand(), 4) np.array()

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

Какие из следующих выражений в Python **корректно** создают словарь (dict)? (Выберите все верные.)

**Варианты:**

{1: 'a', 2: 'b', 3: 'c'}

dict([(1, 'a'), (2, 'b'), (3, 'c')])

{1, 2, 3}

dict(1='a', 2='b')

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

Какие из следующих утверждений о None в Python являются верными?

**Варианты:**

None — это объект единственного экземпляра типа NoneType

None эквивалентен False во всех логических контекстах

None можно использовать как значение по умолчанию для аргумента функции

К переменной, содержащей None, нельзя применить метод .append()

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

Какие из следующих библиотек обычно используются для **визуализации данных** в Python? (Выберите все верные.)

**Варианты:**

matplotlib.pyplot

seaborn

plotly.express

scikit-learn

**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'] &gt; 0]</code>  | А. Замена всех значений меньше 0 на 0 с помощью цикла         |
| 2. <code>df['x'].apply(lambda v: v if v&gt;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'] &lt; 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) | А. 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 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