

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Структурное подразделение «Нефтегазового дела (127)»

УТВЕРЖДЕНА:
на заседании кафедры
Протокол №26 от 10 мая 2025 г.

Рабочая программа дисциплины

«ПРОГРАММИРОВАНИЕ И АЛГОРИТМИЗАЦИЯ»

Специальность: 21.05.06 Нефтегазовые техника и технологии

Разработка и эксплуатация нефтяных и газовых месторождений

Квалификация: Горный инженер (специалист)

Форма обучения: заочная

Документ подписан простой
электронной подписью
Составитель программы:
Молокова Светлана
Васильевна
Дата подписания: 13.06.2026

Документ подписан простой
электронной подписью
Утвердил: Буглов Николай
Александрович
Дата подписания: 17.06.2026

Документ подписан простой
электронной подписью
Согласовал: Шмаков Андрей
Константинович
Дата подписания: 16.06.2026

Год набора – 2026

Иркутск, 2025 г.

1 Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

1.1 Дисциплина «Программирование и алгоритмизация» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ПК-2 Способен выполнять работы по проектированию тех-нологических процес-сов добычи нефти и газа	ПК-2.2

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ПК-2.2	Способны планировать и ставить задачи исследования; планировать этапы исследования, характеризующие методику расчетной и экспериментальной работы с технологическими процессами нефтегазовой отрасли, правильно выбрать программное средства для проведения научных исследований, исходя из требований к профессиональной деятельности	Знать синтаксис и основные конструкции языка Python; типы данных; принципы алгоритмизации; возможности стандартных библиотек для инженерных расчётов (math, numpy, matplotlib). Уметь разрабатывать алгоритмы типовых нефтегазовых задач (расчёт дебита, падение давления, свойства пластовых флюидов); писать и отлаживать программы на Python; визуализировать результаты расчётов. Владеть навыками работы в среде разработки (IDLE/PyCharm); методами численного решения уравнений применительно к задачам разработки месторождений.

2 Место дисциплины в структуре ООП

Изучение дисциплины «Программирование и алгоритмизация» базируется на результатах освоения следующих дисциплин/практик: «Математика», «Физика», «Информационные технологии», «Дополнительные главы по дисциплине "Математика"»

Дисциплина является предшествующей для дисциплин/практик: «Компьютерное моделирование процессов добычи УВС», «Прикладные программные продукты», «Информационные технологии в нефтегазовом комплексе»

3 Объем дисциплины

Объем дисциплины составляет – 3 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)		
	Всего	Учебный год № 4 ый год	Учебный год № 4

		№ 3	
Общая трудоемкость дисциплины	108	36	72
Аудиторные занятия, в том числе:	14	2	12
лекции	8	2	6
лабораторные работы	0	0	0
практические/семинарские занятия	6	0	6
Самостоятельная работа (в т.ч. курсовое проектирование)	90	34	56
Трудоемкость промежуточной аттестации	4	0	4
Вид промежуточной аттестации (итогового контроля по дисциплине)	, Зачет		Зачет

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Учебный год № 3

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)		№	Кол. Час.	
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.			
1	2	3	4	5	6	7	8	9	10	11
1	Основы алгоритмизации. Введение в Python.	1	2					1, 2, 3	20	Устный опрос
2	Управляющие конструкции: ветвления и циклы.							1	4	Устный опрос
	Промежуточная аттестация									
	Всего		2						24	

Учебный год № 4

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)		№	Кол. Час.	
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.			
1	2	3	4	5	6	7	8	9	10	11
1	Функции, работа с данными. Библиотеки numpy, matplotlib.	2	2			2	2	1, 2, 3, 5	19	Собеседование
2	Численные методы в нефтегазовых расчетах на	3	2					1, 2, 3, 5	21	Устный опрос

	Python.								
	Промежуточная аттестация							4	Зачет
	Всего		4			2		44	

4.2 Краткое содержание разделов и тем занятий

Учебный год № 3

№	Тема	Краткое содержание
1	Основы алгоритмизации. Введение в Python.	Понятие алгоритма, свойства, формы записи. Язык Python: установка, среда разработки. Переменные, типы данных (int, float, str, bool). Ввод/вывод данных (print, input). Арифметические операции. Решение задачи: перевод единиц измерения (давление из МПа в кгс/см ²).
2	Управляющие конструкции: ветвления и циклы.	Условный оператор if-elif-else (выбор режима работы скважины). Циклы while и for (расчёт накопленной добычи по месяцам). Операторы break, continue. Строковые методы.

Учебный год № 4

№	Тема	Краткое содержание
1	Функции, работа с данными. Библиотеки numpy, matplotlib.	Списки, кортежи, словари. Определение и вызов функций, аргументы, возврат значений. Работа с библиотекой math. Библиотека numpy: массивы, операции над ними. Библиотека matplotlib: построение графиков (кривая депрессии, падение пластового давления).
2	Численные методы в нефтегазовых расчетах на Python.	Решение нелинейных уравнений (метод половинного деления для расчёта забойного давления). Интерполяция и аппроксимация (обработка результатов гидродинамических исследований (ГДИС)). Численное интегрирование (расчёт объёма жидкости в трубопроводе). Работа с файлами (чтение/запись CSV, Excel).

4.3 Перечень лабораторных работ

Лабораторных работ не предусмотрено

4.4 Перечень практических занятий

Учебный год № 4

№	Темы практических (семинарских) занятий	Кол-во академических часов
1	Решение вычислительной задачи линейной структуры (расчёт забойного давления по формуле Дюпюи).	2
1	Разработка программы табуляции функции и расчёта накопленной добычи с использованием	2

	цикла.	
2	Построение индикаторной линии скважины (P(Q)) с использованием numpy и matplotlib. Оформление графика.	2

4.5 Самостоятельная работа

Учебный год № 3

№	Вид СРС	Кол-во академических часов
1	Контрольная работа для студентов заочной формы обучения	18
2	Подготовка к практическим занятиям (лабораторным работам)	6
3	Проработка разделов теоретического материала	10

Учебный год № 4

№	Вид СРС	Кол-во академических часов
1	Выполнение тренировочных и обучающих тестов	4
2	Оформление отчетов по лабораторным и практическим работам	14
3	Подготовка к зачёту	6
4	Подготовка к практическим занятиям	2
5	Проработка разделов теоретического материала	30

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: «Живое кодирование» (Live Coding) с инженерным кейсом; Работа в микро-группах «Один компьютер — один код»; Мини-соревнование «Кто быстрее рассчитает?»

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по практическим занятиям

Практические занятия (всего 6 академических часов) проводятся в 4-м учебном году.

Цель занятий – закрепить навыки алгоритмизации и программирования на Python применительно к типовым расчётам в нефтегазовой отрасли.

Общие требования к выполнению практических заданий:

Подготовка к занятию – повторить соответствующие разделы теории:

Тема 1 «Основы алгоритмизации. Введение в Python» (переменные, типы данных, ввод/вывод, арифметические операции);

Тема 2 «Управляющие конструкции: ветвления и циклы» (условный оператор if-elif-else, циклы while, for);

Тема 3 «Функции, работа с данными. Библиотеки numpy, matplotlib»;

Тема 4 «Численные методы в нефтегазовых расчётах на Python».

Ход работы на занятии:

Получить индивидуальное задание (вариант) у преподавателя.

Разработать блок-схему или словесный алгоритм решения.
Написать код программы на языке Python в среде IDLE / PyCharm / VS Code.
Выполнить отладку и тестирование на контрольных примерах.
Оформить отчёт (краткий: цель, код, результаты, вывод).
Предъявить результат преподавателю, ответить на контрольные вопросы.
Содержание практических занятий:

№ занятия	Тема	Часы	Формируемый результат
1	Решение вычислительной задачи линейной структуры: расчёт забойного давления по формуле Дюпюи, перевод единиц измерения, ввод/вывод данных.	2	Уметь писать линейные программы с арифметическими операциями и преобразованием типов.
2	Разработка программы табуляции функции и расчёта накопленной добычи нефти по месяцам с использованием циклов и условных операторов.	2	Уметь организовывать циклы, массивы, накапливать суммы, выводить таблицы.
3	Построение индикаторной линии скважины (графика Q(Pзаб)) с использованием библиотек numpy и matplotlib. Оформление графика: подписи осей, легенда.	2	Уметь создавать массивы, строить графики, визуализировать технологические зависимости.

Отчётность:

Отчёт по каждому занятию оформляется в электронном виде (файл .ру и краткий текстовый документ .docx/.pdf) и сдаётся преподавателю не позднее следующего занятия (для заочной формы – в течение сессии).

Защита практической работы – собеседование по коду и контрольным вопросам.

Контрольные вопросы для защиты:

Какие типы данных используются в Python для хранения давлений и дебитов?

Как организовать ветвление для выбора режима работы скважины?

В чём отличие цикла for от while?

Как создать массив numpy и построить график?

Как прочитать данные из CSV-файла с результатами ГДИС?

5.1.2 Методические указания для обучающихся по самостоятельной работе:

Самостоятельная работа (СРС) составляет 90 часов и распределена по двум учебным годам (3-й год – 34 часа, 4-й год – 56 часов). Основные виды СРС: проработка теоретического материала, подготовка к практическим занятиям, выполнение контрольной работы (для заочников), оформление отчётов, подготовка к зачёту.

Общие рекомендации по организации самостоятельной работы:

Планирование времени – распределите объём СРС равномерно в течение семестра. Не откладывайте выполнение заданий на последнюю неделю сессии.

Работа с теорией – изучайте материал по следующей схеме:

прочитайте краткий конспект лекции (выложен в ЭИОС);

посмотрите рекомендованные видеоролики (см. раздел 9 «Ресурсы сети Интернет»);

выполните простые примеры из учебника, запустив их в интерпретаторе Python.

Выполнение практических заданий – каждое практическое занятие требует предварительной подготовки (повторения синтаксиса). Рекомендуется за 1-2 дня до

занятия написать «заготовку» кода по образцу из методички.

Работа над контрольной работой (для 3-го учебного года, 18 часов) – контрольная работа состоит из 3-х задач:

задача 1: линейный расчёт по формуле (пересчёт единиц, расчёт депрессии);

задача 2: табуляция функции с использованием цикла (например, расчёт дебита для ряда забойных давлений);

задача 3: работа с условным оператором (классификация типа флюида по давлению и температуре).

Контрольная работа выполняется в электронном виде (файл .ру + отчёт .docx) и сдаётся на проверку до начала 4-го учебного года.

Оформление отчётов – отчёты по практическим работам должны содержать: титульный лист, цель работы, блок-схему алгоритма, листинг кода, скриншот результата, ответы на контрольные вопросы, вывод. Объём одного отчёта – 2-3 страницы.

Подготовка к зачёту – зачёт проводится в форме собеседования с проверкой работающего кода на ПК. Для подготовки используйте:

типовые оценочные средства (раздел 6.2.2);

свой архив выполненных практических заданий (все 3 работы);

тестовые вопросы, размещённые в ЭИОС.

Рекомендуемый недельный график СРС (на период сессии):

День Вид работы Время (часы)

Понедельник Проработка теории (чтение, конспект) 2

Вторник Решение задач на синтаксис (без ПК, на бумаге)1

Среда Написание программ в среде Python 2

Четверг Оформление отчёта / подготовка к практическому занятию 1

Пятница Работа над контрольной работой (по графику) 2

Контроль самостоятельной работы:

проверка конспектов (выборочно) на практических занятиях;

защита контрольной работы (собеседование по коду);

участие в форуме ЭИОС (решение дополнительных задач, обсуждение ошибок).

Ресурсы для самопроверки:

автоматизированные тесты в системе Moodle (по ссылке от преподавателя);

онлайн-тренажёры: Stepik «Python для начинающих» (уроки 1-5);

сборник задач: Златопольский Д.М. «Сборник задач по программированию» (разделы «Линейные алгоритмы», «Циклы», «Массивы»).

6 Фонд оценочных средств для контроля текущей успеваемости и проведения промежуточной аттестации по дисциплине

6.1 Оценочные средства для проведения текущего контроля

6.1.1 учебный год 3 | Устный опрос

Описание процедуры.

Время и место проведения – опрос проводится в индивидуальном порядке в дни сессии.

Форма проведения – устное собеседование по вопросам, охватывающим теоретический материал и базовые практические навыки. Студент отвечает на 2–3 вопроса из перечня (см. ниже). Допускается демонстрация простого фрагмента кода на доске или ноутбуке.

Подготовка студента – перед опросом студент должен:

изучить конспект лекций и рекомендованную литературу;
выполнить практические задания по теме (написать простые программы);
быть готовым написать короткий код (например, объявить переменные, написать условие if или цикл for) на доске или продиктовать преподавателю.

Порядок проведения:

Преподаватель задаёт вопрос из утверждённого перечня.

Студент даёт развёрнутый ответ (определения, синтаксис, примеры).

По усмотрению преподавателя студенту может быть предложено написать 3–5 строк кода на доске/в тетради или объяснить логику готовой программы.

При затруднениях преподаватель задаёт наводящие вопросы.

Результат фиксируется в ведомости текущей успеваемости.

Фиксация результатов – оценка выставляется в журнал текущего контроля. В случае неудовлетворительного ответа студент обязан пересдать опрос во внеаудиторное время (в течение сессии).

Перечень вопросов для устного опроса

По теме 3-1 «Основы алгоритмизации. Введение в Python»:

Что такое алгоритм? Перечислите основные свойства алгоритма.

Какие формы записи алгоритмов вы знаете? Приведите пример блок-схемы линейного алгоритма.

Какие типы данных существуют в Python? Как объявить переменную?

Чем отличается int от float? Приведите примеры.

Как осуществляется ввод и вывод данных в Python? Напишите простейшую программу, которая запрашивает имя и выводит приветствие.

Какие арифметические операции доступны в Python? Что такое целочисленное деление // и остаток %?

Как преобразовать строку в число? Что произойдёт при вводе букв?

Напишите программу перевода давления из МПа в кгс/см² (1 МПа ≈ 10.197 кгс/см²).

По теме 3-2 «Управляющие конструкции: ветвления и циклы»:

Синтаксис условного оператора if-elif-else. Приведите пример выбора режима работы скважины (фонтанный, газлифтный, ЭЦН).

Как сравнить два числа в Python? Какие операторы сравнения существуют?

Что такое логические операторы and, or, not? Приведите пример из нефтегазовой практики (например, проверка давления и температуры).

Цикл while: синтаксис, пример бесконечного цикла и способ его прерывания.

Цикл for: синтаксис, использование с функцией range(). Как вывести числа от 1 до 10?

Как рассчитать накопленную добычу нефти за 12 месяцев с помощью цикла for?

Напишите фрагмент кода.

Что делают операторы break и continue? Приведите примеры.

В чём разница между циклом for и while? Когда какой лучше использовать

Критерии оценивания.

Оценивание производится по двухуровневой системе («Зачтено» / «Не зачтено»).

Двухуровневая система (зачтено/не зачтено)

Оценка Критерии

Зачтено: Студент правильно отвечает не менее чем на 2 вопроса из 3 (или на 2 из 2).

Демонстрирует понимание базового синтаксиса, может привести пример кода (устно или с записью). Допускаются незначительные ошибки (пропуск двоеточия, путаница в названиях функций), которые исправляет после замечания преподавателя.

Не зачтено: Студент не может ответить ни на один вопрос, либо отвечает только на 1 вопрос из 3 с грубыми ошибками. Не понимает разницы между типами данных, не может написать простейший if или цикл. Отсутствует готовность к практической работе.

6.1.2 учебный год 4 | Собеседование

Описание процедуры.

Собеседование проводится в 4-м учебном году после изучения темы:

4-1. Функции, работа с данными. Библиотеки numpy, matplotlib.

1. Описание процедуры проведения собеседования

Время и место проведения – собеседование проводится индивидуально в дни сессии. На одного студента отводится 8–10 минут.

Форма проведения – индивидуальное собеседование, включающее:

теоретические вопросы (1–2 вопроса) по функциям, структурам данных, библиотекам; демонстрацию работающего кода (студент показывает ранее выполненное практическое задание или пишет небольшой фрагмент кода на ноутбуке/доске); защиту отчёта по практической работе №3 (построение индикаторной линии скважины с использованием numpy и matplotlib).

Подготовка студента – к собеседованию студент должен:

изучить лекционный материал по темам: функции, списки, словари, библиотеки math, numpy, matplotlib;

выполнить практическую работу №3 (код и отчёт);

быть готовым объяснить свой код, ответить на вопросы по синтаксису и логике.

Порядок проведения:

Студент предоставляет преподавателю отчёт по практической работе №3 и открывает свой код в среде разработки (IDLE, PyCharm, VS Code или Google Colab).

Преподаватель задаёт 1–2 теоретических вопроса (см. перечень ниже).

Затем преподаватель просит продемонстрировать работу программы: запустить скрипт, построить график, показать результат расчётов.

По ходу демонстрации студент комментирует ключевые фрагменты кода (создание массива numpy, вызов функции plot, подписи осей).

Преподаватель может задать уточняющие вопросы или попросить внести небольшое изменение в код (например, изменить цвет графика, добавить сетку).

Результат фиксируется в ведомости текущей успеваемости.

Фиксация результатов – оценка выставляется в журнал. В случае «Не зачтено» студент дорабатывает отчёт и код и повторно проходит собеседование в срок до окончания сессии.

2. Перечень вопросов для собеседования

По теме 4-1 «Функции, работа с данными. Библиотеки numpy, matplotlib.»

Что такое функция в Python? Как объявить функцию, передать аргументы и вернуть результат? Приведите пример функции, вычисляющей дебит по формуле Дюпюи.

Чем отличается локальная переменная от глобальной?
Какие встроенные структуры данных для хранения коллекций вы знаете? В чём отличие списка (list) от кортежа (tuple)?
Что такое словарь (dict)? Как получить значение по ключу? Приведите пример хранения параметров скважины (название, дебит, давление).
Для чего предназначена библиотека math? Назовите 2-3 полезные функции для инженерных расчётов.
Что такое numpy? Как создать одномерный массив из списка? Как выполнить поэлементное умножение массивов?
Преимущества массивов numpy перед стандартными списками Python (скорость, удобство операций).
Как построить график с помощью matplotlib.pyplot? Какие функции используются для: создания фигуры (figure); построения линии (plot); подписи осей (xlabel, ylabel); отображения сетки (grid); вывода графика на экран (show)?
Как на одном графике отобразить две кривые (например, дебит и забойное давление)?
Как сохранить график в файл (например, .png)?
Практические задания для демонстрации (на выбор преподавателя):
Написать функцию, которая принимает список забойных давлений и возвращает список дебитов по формуле Дюпюи.
Создать массив numpy из 10 равномерно распределённых чисел от 0 до 30 и построить график синуса (или любой другой функции).
Построить индикаторную диаграмму (ось X – забойное давление, ось Y – дебит) по данным из практической работы №3.

Критерии оценивания.

Зачтено: Студент представил работающий код (практическая работа №3), программа выполняется без синтаксических ошибок, график строится корректно. Студент правильно отвечает не менее чем на 1 теоретический вопрос из 2 (или на 1 из 1) и может прокомментировать свой код (назначение переменных, смысл функций библиотек). Допускаются мелкие недочёты (например, отсутствие подписи осей, неоптимальное оформление), которые студент может исправить по замечанию преподавателя.

Не зачтено: Студент не предоставил рабочий код (программа не запускается, выдаёт ошибки, график отсутствует или построен неверно). Студент не может ответить ни на один теоретический вопрос, не понимает назначения библиотек numpy и matplotlib, не может объяснить логику своей программы.

6.1.3 учебный год 4 | Устный опрос

Описание процедуры.

Устный опрос проводится в 4-м учебном году после изучения темы:

4-2. Численные методы в нефтегазовых расчетах на Python.

1. Описание процедуры проведения устного опроса

Время и место проведения – опрос проводится на практическом или лекционном занятии

в дни сессии, либо индивидуально. На одного студента отводится 5–7 минут.

Форма проведения – устное собеседование по теоретическим вопросам, охватывающим численные методы (решение нелинейных уравнений, интерполяция, аппроксимация, численное интегрирование, работа с файлами). Студент должен продемонстрировать понимание методов и их применения в нефтегазовых расчётах.

Подготовка студента – к опросу студент должен:

изучить лекционный материал и рекомендованные источники по теме;
выполнить практические задания (если предусмотрены) по численным методам;
быть готовым объяснить алгоритм численного метода и его реализацию на Python.

Порядок проведения:

Преподаватель задаёт студенту 2 вопроса из утверждённого перечня (см. ниже).

Студент даёт развёрнутый ответ: суть метода, область применения в нефтегазовом деле, пример реализации на Python (может быть устное описание кода).

При необходимости преподаватель просит написать небольшой фрагмент кода (3–5 строк) на доске/в тетради (например, вызов функции `fsolve`, построение интерполяционного полинома).

По усмотрению преподавателя могут быть заданы уточняющие вопросы.

Результат фиксируется в ведомости текущей успеваемости.

Фиксация результатов – оценка выставляется в журнал. В случае «Не зачтено» студент обязан повторно пройти опрос в срок до окончания сессии.

2. Перечень вопросов для устного опроса

По теме 4-2 «Численные методы в нефтегазовых расчетах на Python»

Какие численные методы используются для решения нелинейных уравнений в нефтегазовых расчётах? Назовите основные методы.

В чём суть метода половинного деления (дихотомии)? Какова его сходимость?

Для какой задачи в нефтегазовом деле может потребоваться решение нелинейного уравнения? Приведите пример (расчёт забойного давления, определение радиуса зоны дренирования и т.п.).

Какая библиотека Python предоставляет готовые функции для решения нелинейных уравнений? Назовите функцию (например, `fsolve` из `scipy.optimize`).

Что такое интерполяция? Чем она отличается от аппроксимации (регрессии)?

Приведите пример обработки результатов гидродинамических исследований (ГДИС) с помощью интерполяции.

Как в Python выполнить интерполяцию табличных данных? Какие функции или библиотеки (например, `scipy.interpolate.interp1d`) вы знаете?

Что такое численное интегрирование? Назовите простейшие методы (метод прямоугольников, трапеций, Симпсона).

Для какой задачи в нефтегазовом деле используется численное интегрирование? (Расчёт объёма жидкости в трубопроводе переменного сечения, определение среднего давления, расчёт накопленной добычи).

Как в Python вычислить определённый интеграл методом трапеций без использования готовых библиотек? Опишите алгоритм.

Какая функция библиотеки `scipy.integrate` позволяет вычислить определённый интеграл? (Например, `quad`).

Как организовать чтение данных из CSV-файла в Python? Какие модули можно

использовать (csv, pandas)?

Как записать результаты расчётов (массив давлений, дебитов) в Excel-файл с помощью библиотеки openpyxl или pandas?

Как построить график по данным, считанным из файла, используя matplotlib?

Критерии оценивания.

Зачтено: Студент правильно отвечает не менее чем на 1 вопрос из 2 (или на 2 вопроса из 3, если задано три). Демонстрирует понимание сути численных методов, может назвать область применения в нефтегазовом деле. Допускаются негрубые ошибки (например, путает названия методов, но правильно объясняет идею; забыл точное имя функции, но знает, для чего она нужна).

Не зачтено: Студент не может ответить ни на один вопрос либо отвечает с грубыми ошибками (не понимает разницы между интерполяцией и аппроксимацией, не может объяснить, зачем решать нелинейное уравнение при расчёте скважины). Отсутствует минимальная подготовка.

6.2 Оценочные средства для проведения промежуточной аттестации

6.2.1 Критерии и средства (методы) оценивания индикаторов достижения компетенции в рамках промежуточной аттестации

Индикатор достижения компетенции	Критерии оценивания	Средства (методы) оценивания промежуточной аттестации
ПК-2.2	«Зачтено»: студент демонстрирует работающую программу для типовой инженерной задачи (объёмом 30-40 строк), может объяснить логику, ответить на 2 вопроса по синтаксису. «Не зачтено»: программа не работает, студент не понимает базовые конструкции.	Собеседование + проверка кода на ПК.

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Учебный год 4, Типовые оценочные средства для проведения зачета по дисциплине

6.2.2.1.1 Описание процедуры

Форма проведения – зачёт проводится в виде собеседования с демонстрацией работающего кода на компьютере. Для студентов заочной формы обучения возможно проведение зачёта в дни сессии индивидуально или в малых группах.

Время проведения – на одного студента отводится 15–20 минут.

Условия допуска к зачёту:

выполнение и защита всех практических работ (3 работы);
положительные результаты текущего контроля (устные опросы и собеседования по темам 3-го и 4-го года);
сдача контрольной работы (для 3-го учебного года).
Структура зачётного задания (билета):

Практическая часть (70% оценки) – написать программу на Python по заданию из билета (объёмом 30–50 строк). Время на написание кода – до 15 минут. Допускается использование любой среды разработки (IDLE, PyCharm, VS Code, Google Colab) и справочных материалов (конспекты, официальная документация). Запрещено копирование готового кода с других устройств.

Теоретическая часть (30% оценки) – ответ на 2–3 вопроса по билету (вопросы охватывают синтаксис Python, алгоритмизацию, библиотеки numpy/matplotlib, численные методы).

Порядок проведения:

Студент получает билет (см. типовой пример ниже).
В течение 10–15 минут пишет программу на компьютере.
Затем запускает программу, демонстрирует результат преподавателю.
Преподаватель задаёт теоретические вопросы (можно во время написания кода или после).
При наличии ошибок в коде преподаватель может предложить студенту исправить их (при этом оценка может быть снижена, но не ниже «Зачтено», если ошибки устранены).
Результат фиксируется в зачётной ведомости.
Фиксация результатов – итоговая оценка «Зачтено» / «Не зачтено» выставляется в ведомость и зачётную книжку.

Пример задания:

Билет №1

Практическая часть:

Написать программу на Python, которая:

Запрашивает у пользователя:
пластовое давление $R_{пл}$ (МПа);
коэффициент продуктивности K ($m^3/(сут \cdot МПа)$);
начальное и конечное забойное давление (МПа) для расчёта.
Создаёт массив забойных давлений от начального до конечного с шагом 1 МПа (использовать `numpy.arange` или `range`).
Для каждого значения забойного давления вычисляет дебит по формуле:
 $Q = K \cdot (R_{пл} - R_{заб})$ если $R_{заб} \leq R_{пл}$, иначе
 $Q = 0$

Выводит на экран таблицу: «Забойное давление (МПа)» – «Дебит ($m^3/сут$)».
Строит график зависимости дебита от забойного давления с помощью `matplotlib`. График должен содержать:
подписи осей («Забойное давление, МПа», «Дебит, $m^3/сут$ »);
заголовок «Индикаторная диаграмма скважины»; сетку.

Теоретические вопросы (ответить устно):

Как объявить функцию в Python? Приведите пример простейшей функции, которая принимает два числа и возвращает их сумму.

Что такое массив numpy? В чём его преимущество перед списком (list)?

Как построить график с помощью библиотеки matplotlib? Перечислите 2-3 основные функции.

Дополнительные пояснения для студента:

Разрешено пользоваться любыми справочными материалами (конспекты, документация).

Запрещено копировать код из Интернета или с мобильного телефона.

Код должен быть написан самостоятельно.

После выполнения необходимо показать работающую программу преподавателю.

Билет №2 (альтернативный пример, для полноты фонда)

Практическая часть:

Написать программу, которая:

Считывает из файла data.csv два столбца: «Забойное давление (МПа)» и «Дебит (м³/сут)» (файл должен быть создан заранее с 5–6 строками данных).

На основе считанных данных строит точки на графике (scatter) и аппроксимирующую прямую (линейную регрессию) с помощью numpy.polyfit.

Выводит уравнение прямой (например, « $Q = a \cdot P_{заб} + b$ ») на экран.

По запросу пользователя рассчитывает дебит для введённого забойного давления по полученной формуле.

Теоретические вопросы:

Что такое интерполяция? Чем она отличается от аппроксимации?

Как выполнить чтение CSV-файла в Python (назовите модуль и основные функции)?

Как создать словарь (dict) и получить значение по ключу?_

6.2.2.1.2 Критерии оценивания

Зачтено	Не зачтено
<p>Практическая часть: Программа запускается без синтаксических ошибок, корректно решает поставленную задачу (расчёт, таблица, график – согласно заданию). Переменные названы осмысленно, код структурирован (использованы функции, если требуется). Допускаются 1–2 логические ошибки, которые студент исправляет после замечания преподавателя.</p> <p>Теоретическая часть: Студент правильно отвечает на 2 вопроса из 3 (или на 2 из 2). Ответы показывают</p>	<p>Практическая часть (код): Программа не запускается (ошибки синтаксиса), либо запускается, но выдаёт неверные результаты из-за грубых логических ошибок. Студент не может исправить ошибки.</p> <p>Теоретическая часть (ответы на вопросы): Студент не может ответить ни на один вопрос, либо отвечает только на 1 вопрос с грубыми ошибками. Студент не понимает собственного кода, не может объяснить, что делает программа.</p>

<p>понимание материала (синтаксис, работа с библиотеками, численные методы). Допускаются неточности, не искажающие сути. Студент уверенно ориентируется в коде, объясняет его логику. Итоговая оценка «Зачтено» ставится при выполнении всех трёх критериев (практическая часть – «Зачтено», теоретическая часть – «Зачтено», общее впечатление положительное).</p>	
--	--

7 Основная учебная литература

1. Казанцев Т. Искусственный интеллект и машинное обучение. Основы программирования на Python / Т. Казанцев, 2020. - 170.

[Сайт] – URL: <https://www.litres.ru/book/timur-kazancev/iskusstvennyu-intellekt-i-mashinnoe-obuchenie-osnovy-progra-59162107/>

8 Дополнительная учебная литература и справочная

1. Незнанов А. А. Программирование и алгоритмизация : учебник для вузов по направлению подготовки "Автоматизированные технологии и производства" / А. А. Незнанов, 2010. - 303.

2. Программирование алгоритмов защиты информации : учеб. пособие / А. В. Домашев [и др.], 2000. - 279.

9 Ресурсы сети Интернет

1. <http://library.istu.edu/>
2. <https://e.lanbook.com/>

10 Профессиональные базы данных

1. <http://new.fips.ru/>
2. <http://www1.fips.ru/>

11 Перечень информационных технологий, лицензионных и свободно распространяемых специализированных программных средств, информационных справочных систем

1. Лицензионное программное обеспечение Системное программное обеспечение
2. Лицензионное программное обеспечение Пакет прикладных офисных программ
3. Лицензионное программное обеспечение Интернет-браузер
4. Свободно распространяемое программное обеспечение Python

12 Материально-техническое обеспечение дисциплины

1. Учебная аудитория для проведения лекционных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Оснащение: комплект учебной мебели, рабочее место преподавателя, доска. Мультимедийное оборудование (в том числе переносное): мультимедийный проектор, экран, акустическая система, компьютер с выходом в интернет.

2. Учебная аудитория для проведения лабораторных/практических (семинарских) занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Оснащение: комплект учебной мебели, рабочее место преподавателя, доска. Мультимедийное оборудование (в том числе переносное): мультимедийный проектор, экран, акустическая система, компьютер с выходом в интернет.