МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Министерство образования Свердловской области

управление образования Березовского муниципального округа
БМАОУ СОШ №21

СОГЛАСОВАНО	УТВЕРЖДЕНО
председатель педагогического совета	директор БМАОУ СОШ №21
Кулешова Н.М.	Щупова И.Ю.
Протокол № 14	Приказ №59/1-о
От «29» июня 2026 г.	от «29» июня 2026 г.

Дополнительная общеобразовательная общеразвивающая программа

Программирование на Python

Техническая направленность

Уровень: базовый

Возраст обучающихся: 12-16 лет

Срок реализации: один год, 2026/27 учебный год

Объём: 68 академических часов, 34 учебные недели

Форма обучения: очная

Березовский муниципальный округ
пос. Лосиный
2026

Оглавление

Разделы программы и приложенные материалы

1. Пояснительная записка 3
2. Учебный план 4
3. Календарный учебный график 5
4. Содержание программы 12
5. Воспитание 15
6. Планируемые результаты 16
7. Ресурсное обеспечение 17
8. Список литературы и электронных ресурсов 19
9. Оценочные материалы 20
10. Методические материалы 22

Оценочные и методические материалы входят в состав программы. Часы диагностики, промежуточной и итоговой работы включены в общий объём, дополнительные занятия сверх учебного плана не предполагаются.

1. Пояснительная записка

1.1. Назначение и особенности программы

Python позволяет связать алгоритм с текстом программы, данными и проверяемым результатом. Начинающий разработчик сталкивается не только с синтаксисом, но и с вопросами о допустимом вводе, пустых наборах, повторном использовании кода и сохранении файлов. Программа вводит эти вопросы постепенно на небольших консольных задачах, доступных подросткам без профессиональной подготовки.

Базовый курс ограничен ядром языка и стандартной библиотекой: значения, ввод и вывод, ветвления, циклы, строки, списки, словари, функции, текстовые файлы и обработка ожидаемых ошибок. Веб-сервисы, машинное обучение, сетевые боты и сложные библиотеки не включаются ради формального расширения тем. Это оставляет время на самостоятельное чтение кода, отладку и тестирование.

Все наборы данных учебные и неперсональные. Итоговое приложение работает локально и не требует регистрации, ключей доступа или загрузки данных во внешние сервисы. Ученик создаёт небольшой анализатор измерений с меню и сохранением результатов, документирует правила ввода и подтверждает корректность тестами. Обучение не обещает профессиональной квалификации, но даёт прочную основу для дальнейшего программирования.

1.2. Адресат и условия зачисления

Возраст обучающихся: 12-16 лет. Уровень программы: базовый.

Предполагаются базовая работа с файлами, чтение коротких инструкций, арифметические действия и понимание последовательности команд. Опыт Scratch полезен, но не обязателен. Входная работа выявляет необходимые опоры; предварительное знание синтаксиса Python не требуется. При отсутствии навыков управления файлами предоставляется вводная практическая карточка.

Рекомендуемый состав группы: 12-15 человек; предпочтительно индивидуальные компьютеры. Допустимы 8 мест: при 15 детях семь пар и один участник индивидуально, с ротацией ролей. Каждый сохраняет личный код и выполняет контрольное изменение. Педагог просматривает короткие программы, не только итоговый экран.

1.3. Цель и задачи

Цель: Научить самостоятельно разрабатывать, объяснять и тестировать небольшую локальную программу Python, которая обрабатывает структурированные учебные данные и корректно реагирует на ожидаемые ошибки ввода.

Обучающие задачи

• Освоить запуск Python, редактирование файлов .py и чтение диагностических сообщений.

• Применять числовые и строковые типы, преобразования, форматированный ввод и вывод.

• Реализовывать ветвления и циклы с корректными граничными условиями.

• Обрабатывать списки и словари, выделять повторяемые действия в функции.

• Читать и сохранять учебные файлы, обрабатывать ожидаемые ошибки без потери исходных данных.

• Создавать набор обычных, граничных и ошибочных тестов и подтверждать исправление повторным запуском.

Развивающие задачи

• Разделять задачу на входные данные, обработку и результат.

• Трассировать состояние переменных и объяснять причину ошибки.

• Планировать функции и интерфейсы до написания всего приложения.

• Проводить предметную проверку чужого кода и учитывать замечания.

Воспитательные задачи

• Ответственно относиться к данным и не запускать непонятный код.

• Уважать авторство и указывать использованные фрагменты.

• Сохранять настойчивость в отладке без сокрытия неудачных тестов.

1.4. Объём и организация занятий

Программа рассчитана на один год: 34 учебные недели, 68 академических часов. Проводятся два занятия в неделю в разные дни, каждое продолжительностью 40 минут. Такая продолжительность академического часа выбрана для данной программы и не обозначает единого федерального норматива. Расписание, перерывы и пределы непрерывной работы с экраном закрепляются в локальном режиме с учётом возраста и действующих санитарных требований. Работа с экраном чередуется с планированием на бумаге, обсуждением, сменой позы и практикой без экрана. Учебные часы не означают непрерывную работу за компьютером или с инструментом. Обязательные домашние задания не предусмотрены.

Годовой объём составляет 68 академических часов за 34 недели, два отдельных занятия еженедельно. Входная, промежуточная и итоговая проверки включены в план. Обязательных домашних заданий нет. Экранные периоды чередуются с бумажной трассировкой, обсуждением алгоритма и планированием тестов; режим определяется организацией по действующим возрастным требованиям.

Рабочий цикл: прочитать условие, назвать вход и выход, предсказать пример, написать минимальный фрагмент, проверить и объяснить. Педагог последовательно уменьшает объём стартового кода. В паре роли автора и рецензента меняются, контрольная работа остаётся индивидуально объясняемой. После пропуска предоставляются краткая справка, небольшой исправляемый пример и обязательная личная проба, связывающая новую тему с уже изученной.

Фактические даты начала и окончания, каникулы, праздничные дни, расписание и переносы организация указывает в своём календаре до начала реализации. Нумерация недель в программе отражает последовательность обучения и не задаёт несуществующих календарных дат.

1.5. Основания разработки

Программа разработана с учётом Федерального закона № 273-ФЗ и Порядка, утверждённого приказом Минпросвещения России № 629. При выборе структуры использованы методические рекомендации Минпросвещения России от 23.01.2026 № АБ-254/06. Организация самостоятельно утверждает программу и формы контроля в пределах действующих требований и локальных актов. Программа не предполагает освоения профессиональной квалификации.

При реализации учитываются СП 2.4.2.4283-26 и применимые гигиенические нормативы. До 01.03.2027 необходимо уточнить санитарную нормативную базу для оставшейся части учебного года в связи с установленным сроком действия СанПиН 1.2.3685-21. Ссылки на документы приведены в разделе 8.

2. Учебный план

Теория включает объяснение, демонстрацию и разбор принципов. Практика включает решение задач, изготовление, настройку, испытание, обсуждение собственных результатов и оценочные работы. В колонках «Т» и «П» указаны часы теории и практики. Один и тот же час учитывается только в одной колонке.

№	Раздел	Всего	Т	П	Форма контроля
1	Среда разработки и алгоритмическая запись	4	2	2	Файл запускается повторно, ошибка локализуется по строке сообщения.
2	Значения типы и пользовательский ввод	10	3	7	Типы выбраны верно, целочисленное деление и остаток объяснены.
3	Ветвления циклы и граничные случаи	12	3	9	Границы проверены, цикл завершается, накопитель и счётчик различены.
4	Коллекции функции и промежуточная проверка	12	3	9	Пустой набор обработан, функция возвращает результат и не зависит от случайного внешнего состояния.
5	Текст файлы и ожидаемые ошибки	10	2	8	Ошибочный ввод даёт понятное сообщение, исходный файл остаётся сохранным.
6	Структура приложения и тестирование	6	1	5	Тесты запускаются повторно, ошибка исправлена без нарушения прежних случаев.
7	Локальное приложение Лаборатория измерений	10	1	9	Основные команды работают, недопустимый ввод не повреждает данные, результаты воспроизводимы.
8	Защита программы и повторная проверка	4	0	4	Автор объясняет функцию, тест и обоснованное изменение программы.
	Итого	68	15	53	Контроль в составе указанных часов

3. Календарный учебный график

Учебный период: 2026/27 год. Продолжительность: 34 учебные недели. В колонке «№» указан номер учебной недели. В каждой строке запланированы два разных занятия по одному академическому часу. Т/П обозначает количество часов теории и практики за неделю; сумма в строке равна двум.

№	Тема	Содержание двух занятий	Т/П	Контроль
1	Алгоритм и запуск	Занятие 1. Разобрать задачу расчёта учебных комплектов, выделить вход, обработку и выход, выполнить входные задания и представить решение короткой бумажной последовательностью действий. Занятие 2. Создать рабочую папку и файл Python, вывести текст и выражение, запустить скрипт, остановить выполнение и повторно открыть сохранённый исходный код.	1/1	Вход и выход названы, файл запускается из своей папки.

№	Тема	Содержание двух занятий	Т/П	Контроль
2	Сообщение об ошибке	Занятие 1. Сравнить интерактивную консоль и запуск файла, рассмотреть сообщение о пропущенной кавычке, определить номер строки и отличие синтаксиса от результата. Занятие 2. Исправить два подготовленных коротких примера, повторить запуск, записать причину ошибки и сохранить рабочую версию с понятным именем файла проекта.	1/1	Ошибка найдена по сообщению, причина названа без случайной переписи.
3	Переменная и присваивание	Занятие 1. Проследить изменение переменной после последовательных присваиваний, отличить имя от значения, предсказать вывод и заполнить бумажную таблицу состояний программы. Занятие 2. Написать расчёт стоимости одинаковых условных деталей, использовать осмысленные имена, проверить три набора чисел и сопоставить вывод с ручным вычислением.	1/1	Три результата верны, присваивание объяснено как изменение связанного значения.
4	Арифметические операции	Занятие 1. Исследовать сложение, умножение, деление и скобки в коротких выражениях, сравнить прогноз с интерпретатором и отметить порядок выполнения арифметических действий. Занятие 2. Создать конвертер сантиметров в метры, проверить ноль и дробный результат, добавить единицы измерения и записать допустимый формат исходного значения.	0/2	Порядок операций верен, единицы и допустимый ввод указаны.
5	Ввод и тип	Занятие 1. Объяснить строковый результат input, сравнить текстовые и числовые операции, выбрать преобразование int или float и предсказать результат короткого примера. Занятие 2. Добавить пользовательский ввод к калькулятору, исправить сложение строк вместо чисел, проверить три корректных значения и обозначить пока не обработанный неверный ввод.	1/1	Тип преобразован осмысленно, ограничение текущей версии обозначено честно.
6	Целые наборы и остаток	Занятие 1. Разделить заданное число деталей на комплекты фиксированного размера, предсказать частное и остаток, проверить вычисления операциями целочисленного деления и остатка. Занятие 2. Написать программу для семнадцати деталей по пять в наборе, получить три комплекта и две оставшиеся детали, затем проверить граничный случай.	0/2	Для 17 и 5 получены 3 и 2, граница проверена.

№	Тема	Содержание двух занятий	Т/П	Контроль
7	Понятный вывод	Занятие 1. Рассмотреть форматированную строку, выбрать подписи и единицы, отличить округление отображения от изменения вычисления и подготовить образец читаемого отчёта пользователя. Занятие 2. Объединить ввод, расчёт и вывод небольшого конвертера, проверить три сценария, дать партнёру прочитать результат и исправить непонятную подсказку программы.	1/1	Пользователь понимает ввод и результат, формат не скрывает ошибку.
8	Сравнения и ветвление	Занятие 1. Сопоставить присваивание и сравнение, определить истинность условий, нарисовать две ветви решения и предсказать результат для значения на заданной границе. Занятие 2. Создать классификатор значения относительно порога десять, проверить девять, десять и одиннадцать, исправить неверный знак сравнения и записать три ожидаемых ответа.	1/1	Граничные значения обработаны верно, сравнение не заменено присваиванием.
9	Несколько диапазонов	Занятие 1. Спланировать три непересекающихся диапазона учебного показателя, выбрать порядок if, elif и else, проверить полноту покрытия на бумажной числовой прямой. Занятие 2. Реализовать классификатор, проверить обе границы и промежуточное значение, найти недостижимую ветвь в подготовленном примере и объяснить причину ошибки порядка условий.	0/2	Диапазоны не пересекаются, каждая ветвь достижима и проверена.
10	Цикл for и range	Занятие 1. Рассмотреть последовательность range, предсказать первые и последние значения, отметить невключённую правую границу и составить план вывода таблицы кратных чисел. Занятие 2. Написать цикл вывода пяти кратных трём, получить 3, 6, 9, 12, 15, исправить ошибку одного шага и проверить минимальное число повторов.	1/1	Правая граница учтена, пять кратных выведены без лишнего элемента.
11	Счётчик и накопитель	Занятие 1. Проследить сумму и количество по короткому набору, разделить назначение двух переменных, записать начальные значения и место обновления внутри цикла. Занятие 2. Реализовать сумму чисел от одного до пяти и подсчёт подходящих значений, проверить результаты пятнадцать и ожидаемое количество по ручному расчёту.	0/2	Счётчик и сумма обновляются правильно, ручной результат совпадает.

№	Тема	Содержание двух занятий	Т/П	Контроль
12	Цикл while	Занятие 1. Рассмотреть условие продолжения, найти действие, меняющее его истинность, предсказать завершение и обозначить риск бесконечного выполнения на бумажной трассировке программы. Занятие 2. Создать меню с командой выхода, проверить повторение и завершение, остановить подготовленный ошибочный цикл штатной кнопкой и исправить отсутствующее обновление состояния.	1/1	Меню повторяется и завершается, причина бесконечного цикла объяснена.
13	Поиск числа	Занятие 1. Составить алгоритм угадывания заданного числа с ограничением попыток, определить условия меньшего, большего и точного ответа, подготовить набор контрольных последовательностей. Занятие 2. Реализовать программу, проверить успех с первой попытки, последнюю допустимую попытку и исчерпание лимита, записать результаты без случайного выбора секретного числа.	0/2	Три сценария завершения корректны, граница числа попыток соблюдена.
14	Список и индекс	Занятие 1. Рассмотреть список измерений, определить длину и индексы, предсказать обращение к первому и последнему элементам, обсудить недопустимый индекс и пустой набор. Занятие 2. Создать список, добавить значение, обойти элементы циклом и вывести пронумерованный отчёт, проверяя отличие номера для пользователя от индекса в коде.	1/1	Индексы начинаются с нуля, вывод понятен, пустой набор замечен.
15	Анализ набора	Занятие 1. Вычислить количество, сумму, минимум и максимум заданных измерений, проверить результат вручную и определить, какие операции нельзя выполнять над пустым списком. Занятие 2. Написать анализатор, добавить отдельную ветвь пустого набора, проверить одно значение, повторы и обычный список, сохранив все ожидаемые и фактические результаты.	0/2	Пустой набор не вызывает ошибку, статистика верна для повторов.
16	Фильтрация и словарь	Занятие 1. Рассмотреть отбор значений не ниже порога, объяснить создание нового списка, затем описать учебный образец словарём с названием и числовым измерением. Занятие 2. Реализовать фильтр без изменения исходного списка во время обхода, создать две словарные записи и проверить доступ к значениям по понятным ключам.	1/1	Исходный список сохранён, граница фильтра включена, ключи объяснены.

№	Тема	Содержание двух занятий	Т/П	Контроль
17	Промежуточная проверка	Занятие 1. Обработать список 12, 9, 15, 9, 20: вывести количество, сумму, среднее, минимум, максимум и число значений не ниже двенадцати. Занятие 2. Проверить пустой список и список из одного нуля, исправить найденную ошибку, объяснить цикл и предъявить таблицу ожидаемых результатов контрольной задачи.	0/2	Получены 5, 65, 13, 9, 20 и 3, пустота обработана.
18	Функция и возврат	Занятие 1. Разделить повторяемый анализ на функцию, определить параметры и возвращаемое значение, сопоставить return с печатью и проследить простой вызов на бумаге. Занятие 2. Написать функцию суммы или статистики, вызвать с двумя наборами, использовать результат в разных отчётах и устранить зависимость от внешней случайной переменной.	1/1	Функция возвращает результат, два вызова работают независимо друг от друга.
19	Список записей	Занятие 1. Создать список словарей условных образцов, определить одинаковые ключи и допустимые значения, написать функцию выбора записей по числовому порогу измерения. Занятие 2. Проверить функцию на пустом наборе, совпадении с порогом и повторяющихся измерениях, вывести результат и объяснить, почему порядок исходных записей сохраняется.	0/2	Фильтр работает на границе, структура записей единообразна и объяснена.
20	Строки и подготовка текста	Занятие 1. Рассмотреть пробелы, регистр и разбиение строки, предсказать результаты strip и split, определить, какие изменения допустимы для учебной подписи образца. Занятие 2. Написать нормализацию ввода команды меню, проверить лишние пробелы и разные регистры, сохранить исходную подпись образца и объяснить различие обработки команд и данных.	1/1	Команды нормализованы, значимые данные не изменяются без необходимости.
21	Текстовый файл	Занятие 1. Создать учебный текстовый файл, прочитать строки через with с UTF-8, проверить перевод строки и записать количество реально прочитанных записей набора. Занятие 2. Сохранить отдельный отчёт новым именем, повторно открыть его, проверить русские подписи и убедиться, что исходный файл остался неизменным после работы.	0/2	Файл читается повторно, кодировка указана, оригинал не перезаписан.

№	Тема	Содержание двух занятий	Т/П	Контроль
22	CSV и структура	Занятие 1. Рассмотреть заголовок name,value и строки учебного CSV, определить разделитель, типы и допустимый диапазон, составить план проверки записи перед добавлением. Занятие 2. Прочитать таблицу модулем csv, преобразовать числовое поле, вывести записи и сравнить результат с исходным файлом, обнаруживая пропущенную или лишнюю строку набора.	1/1	Заголовок и типы проверены, все исходные записи учтены.
23	Ожидаемые ошибки	Занятие 1. Добавить обработку ValueError для нечислового ввода и FileNotFoundError для отсутствующего файла, подготовить понятные сообщения без скрытия других ошибок программы. Занятие 2. Проверить пустую строку, буквы и неверный путь, убедиться, что текущие данные сохраняются, затем повторить корректный ввод и успешную загрузку файла.	0/2	Ошибки объяснены пользователю, данные сохранны, корректный путь продолжает работать.
24	Круг сохранения	Занятие 1. Сохранить учебный список записей в новый CSV, повторно загрузить его, сопоставить названия и значения и проверить документированный десятичный разделитель ввода. Занятие 2. Добавить проверку значения от нуля до тысячи, испытать обе границы и выход за диапазон, уточнить инструкцию без изменения исходного учебного набора.	0/2	Сохранение воспроиз-водимо, границы приняты, недопустимые значения отклонены корректно.
25	Структура приложения	Занятие 1. Разделить длинный скрипт на ввод, обработку и вывод, определить функции без лишних побочных действий и составить схему вызовов будущего приложения. Занятие 2. Перенести вычисления в отдельные функции, проверить прежние примеры и написать первые assert для заранее известных результатов, не используя их вместо проверки пользовательского ввода.	1/1	Функции разделены, прежние результаты сохранены, тесты выполняются отдельно.
26	Граничные тесты	Занятие 1. Подготовить случаи пустого списка, одного нуля, повторов, равенства порогу и обычного набора, записать ожидаемые результаты до запуска проверяемой функции. Занятие 2. Запустить тесты на подготов-ленных ошибках, исправить неверную границу и деление на ноль, повторить весь набор и сохранить журнал найденных причин.	0/2	Граничные случаи представлены, исправления подтверждены полным повторным набором.

№	Тема	Содержание двух занятий	Т/П	Контроль
27	Рецензирование кода	Занятие 1. Передать партнёру функции и тесты, проверить имена, повторяющийся код и воспроизводимость запуска, сформулировать замечание с конкретным входом и ожидаемым результатом. Занятие 2. Разобрать замечания, принять обоснованное исправление или объяснить отказ, повторить тесты и подготовить личный план реализации итогового приложения с меню и файлами.	0/2	Замечания доказательны, решение объяснено, тесты проходят после изменения.
28	Задание Лаборатории измерений	Занятие 1. Определить запись name,value, команды меню, диапазон от нуля до тысячи и правила ошибки, составить схему функций и таблицу восьми будущих тестов. Занятие 2. Подготовить базовый CSV с вымышленными образцами, реализовать хранение и просмотр списка, проверить пустой набор и сохранить первую работающую версию приложения.	1/1	Структура и правила определены, просмотр работает на пустом наборе.
29	Добавление и статистика	Занятие 1. Реализовать проверяемое добавление записи, запретить пустое название и недопустимое значение, проверить ноль, границы диапазона и нечисловую строку пользовательского ввода. Занятие 2. Добавить статистику и фильтр по порогу, проверить базовый набор со значениями 12, 9, 15, 4 и сравнить результат с ручным вычислением.	0/2	База даёт сумму 40 и среднее 10, ошибки не меняют данные.
30	Файлы и меню проекта	Занятие 1. Соединить функции меню, загрузки и сохранения CSV, обеспечить понятный выход, проверить отсутствие файла и сохранение текущего списка при ошибке чтения. Занятие 2. Выполнить полный круг добавления, сохранения и повторной загрузки, сравнить записи, оформить сообщения об ошибках и отметить недостающую проверку в журнале проекта.	0/2	Меню устойчиво, круг файла сохраняет данные, отсутствующий файл обработан.
31	Полный набор тестов	Занятие 1. Выполнить минимум восемь заранее подготовленных тестов, записать факт и ожидание, локализовать наиболее существенный сбой и исправить его минимальным изменением функции. Занятие 2. Повторить весь набор после исправления, провести пользовательский запуск по инструкции и уточнить непонятную подсказку или правило допустимого ввода собственного приложения.	0/2	Восемь тестов выполнены повторно, изменение не нарушило прежние случаи.

№	Тема	Содержание двух занятий	Т/П	Контроль
32	Документация и комплект	Занятие 1. Подготовить инструкцию запуска, описание команд, структуру CSV, ограничения и источники фрагментов, сохранить код, тесты и учебные данные в одной проектной папке. Занятие 2. Запустить проект из чистой копии папки, проверить комплектность, отрепетировать объяснение функции и подготовить небольшое изменение порога или формата отчёта программы.	0/2	Комплект запускается, инструкция достаточна, автор объясняет расчёт.
33	Итоговая защита	Занятие 1. Показать добавление, статистику, фильтр и сохранение на учебных данных, продемонстрировать ошибочный ввод и предъявить результаты обычных и граничных тестов приложения. Занятие 2. Ответить на вопросы о типах, цикле и возврате функции, выполнить небольшое изменение по заданию и определить недостающее действие для повторной проверки.	0/2	Личное понимание подтверждено объяснением, изменением и повторным запуском теста.
34	Повторная проверка и итог	Занятие 1. Подтвердить недостающее умение после обратной связи, обновить документацию и показать другому пользователю локальную программу с честно обозначенными ограничениями текущей версии. Занятие 2. Сохранить финальный комплект и резервную копию по правилам организации, выбрать фрагмент кода в портфолио и сформулировать следующий вопрос для самостоятельного изучения.	0/2	Финальный комплект сохранён, результат подтверждён, следующий вопрос конкретен.

Всего по графику: 68 часов; теория 15, практика 53. Промежуточная работа предусмотрена к 17-й неделе; итоговая работа завершает обучение.

4. Содержание программы

4.1. Среда разработки и алгоритмическая запись

Объём: 4 ч. Теория: 2 ч. Практика: 2 ч.

Теоретическое содержание

Интерпретатор выполняет программу, редактор хранит её текст, консоль показывает ввод и вывод. Интерактивный режим отличается от запуска файла. Алгоритм описывает последовательность до выбора синтаксиса. Ошибка синтаксиса мешает прочитать код, ошибка выполнения возникает при действии, логическая ошибка даёт неправильный результат без сообщения интерпретатора. Рабочая папка ограничивает учебные файлы.

Практическая работа

Участники выполняют входные задания, создают папку и первый файл, выводят текст и результат арифметического выражения. Сравнивают консольную команду и сохранённый скрипт, повторно открывают файл. Намеренно исправляют пропущенную кавычку и читают

номер строки. На бумаге описывают алгоритм расчёта стоимости одинаковых учебных деталей. Педагог проверяет остановку зависшего учебного процесса штатной кнопкой среды.

Продукт работы: Первый скрипт и схема «вход, обработка, выход».

Контроль: Файл запускается повторно, ошибка локализуется по строке сообщения.

4.2. Значения типы и пользовательский ввод

Объём: 10 ч. Теория: 3 ч. Практика: 7 ч.

Теоретическое содержание

Переменная связывает имя со значением, присваивание не является математическим равенством. Целые числа, числа с дробной частью, строки и логические значения имеют разные операции. input возвращает строку, поэтому числовой ввод требует преобразования. Операции /, // и % различаются. Форматированный вывод делает результат понятным; округление отображения не исправляет неверный алгоритм.

Практическая работа

Дети пишут калькулятор количества комплектов, конвертер длины и программу разделения предметов на полные наборы и остаток. Предсказывают результат смешения строк и чисел, исправляют преобразование. Используют осмысленные имена, f-строки и единицы измерения. Сохраняют три контрольных примера для каждой программы. Ввод пока предполагается числовым по условию; полноценная обработка ошибочной строки вводится позже, и это ограничение явно записывается.

Продукт работы: Три короткие программы с понятными подсказками и контрольными примерами.

Контроль: Типы выбраны верно, целочисленное деление и остаток объяснены.

4.3. Ветвления циклы и граничные случаи

Объём: 12 ч. Теория: 3 ч. Практика: 9 ч.

Теоретическое содержание

Условие вычисляется как истинное или ложное; сравнение отличается от присваивания. if, elif и else задают взаимоисключающие ветви, логические операции объединяют проверки. Цикл for перебирает элементы или range, правая граница range не включается. while продолжается до изменения условия; необходимо обеспечить завершение. Счётчик и накопитель обновляются в правильном месте.

Практическая работа

Участники создают классификатор учебного значения по диапазонам и проверяют точки на границе. Пишут таблицу кратных, сумму чисел и подсчёт значений, удовлетворяющих условию. Выполняют трассировку переменных по строкам. Создают меню с командой выхода и ограниченную игру угадывания заданного числа без внешних сервисов. Находят ошибку «на один шаг» и бесконечный цикл в подготовленном примере. Каждое исправление проверяют на минимальном и обычном входе.

Продукт работы: Программы классификации, накопления и меню с корректным завершением.

Контроль: Границы проверены, цикл завершается, накопитель и счётчик различены.

4.4. Коллекции функции и промежуточная проверка

Объём: 12 ч. Теория: 3 ч. Практика: 9 ч.

Теоретическое содержание

Список хранит последовательность элементов, индексы начинаются с нуля; пустой список требует отдельного решения. Словарь связывает ключи и значения. Изменение коллекции во время обхода может нарушить замысел, поэтому сначала формируется новый результат. Функция имеет параметры и возвращаемое значение; print показывает текст, return передаёт результат вызывающему коду. Локальные переменные снижают скрытые зависимости.

Практическая работа

Дети анализируют список измерений: считают количество, сумму, минимум и максимум, фильтруют значения. На неделе 17 выполняют самостоятельную проверку на заданном наборе и пустом списке. Затем выделяют анализ в функцию, сравнивают возврат значения и печать. Создают словарь условного образца с названием и измерением, объединяют записи в список. Проверяют функцию отдельными вызовами и используют результат в двух разных вариантах вывода.

Продукт работы: Анализатор списка и набор функций для учебных записей.

Контроль: Пустой набор обработан, функция возвращает результат и не зависит от случайного внешнего состояния.

4.5. Текст файлы и ожидаемые ошибки

Объём: 10 ч. Теория: 2 ч. Практика: 8 ч.

Теоретическое содержание

Строка является последовательностью символов; пробелы и регистр могут влиять на сравнение. Файл имеет путь, режим открытия и кодировку. Конструкция with закрывает ресурс после работы. CSV представляет табличные записи, но сам формат не проверяет допустимость значений. Исключения ValueError и FileNotFoundError обрабатываются там, где ожидаемы; перехват всех ошибок без объяснения скрывает дефекты.

Практическая работа

Участники очищают учебные подписи от лишних пробелов, читают строки локального файла и сохраняют отдельный отчёт. Через модуль csv загружают вымышленные записи, проверяют заголовки и преобразуют значения. Добавляют обработку пустого ввода, нечисловой строки и отсутствующего файла. Исходный набор не перезаписывают: результат сохраняют новым именем. Проверяют повторное чтение и русские подписи в UTF-8, документируют допустимый десятичный разделитель.

Продукт работы: Чтение CSV, проверка входа и сохранение отчёта без изменения оригинала.

Контроль: Ошибочный ввод даёт понятное сообщение, исходный файл остаётся сохранным.

4.6. Структура приложения и тестирование

Объём: 6 ч. Теория: 1 ч. Практика: 5 ч.

Теоретическое содержание

Приложение разделяется на ввод, обработку данных и представление результата. Чистую вычислительную функцию легче проверить, чем код с постоянным запросом ввода. Тест связывает вход с ожидаемым выходом; обычные, граничные и ошибочные случаи дополняют друг друга. assert подходит для учебных проверок, но не заменяет обработку пользовательского ввода. Регрессионная проверка повторяет старые тесты после изменения.

Практическая работа

Пары реорганизуют длинный скрипт в несколько функций с понятными именами. Для анализа списка задают тесты пустого набора, одного значения, повторов, нуля и границы фильтра. Исправляют три подготовленные ошибки: неверную границу, деление на ноль и потерю строки при чтении. Рецензент отмечает только воспроизводимые дефекты и непонятные имена. Автор принимает или отклоняет замечание с объяснением и запускает весь набор проверок.

Продукт работы: Модуль функций, отдельный файл тестов и журнал исправлений.

Контроль: Тесты запускаются повторно, ошибка исправлена без нарушения прежних случаев.

4.7. Локальное приложение Лаборатория измерений

Объём: 10 ч. Теория: 1 ч. Практика: 9 ч.

Теоретическое содержание

Техническое задание определяет структуру записи, допустимые значения, команды меню и поведение при ошибке. Список словарей подходит для небольшого учебного набора. Проверка ввода должна предшествовать изменению данных. Сохранение и загрузка рассматриваются как отдельные операции, результаты которых проверяются. Документация сообщает требования запуска и ограничения, не обещая промышленной надёжности.

Практическая работа

Участник создаёт консольное приложение для вымышленных образцов с названием и числовым измерением от 0 до 1000. Реализует добавление, просмотр, статистику, фильтр по порогу, сохранение и загрузку CSV, выход. Вычислительная логика выделяется минимум в три функции. Автор проверяет базовый набор, пустые данные, ноль, границу, неверный ввод, отсутствие файла и повторное чтение. После доработки передаёт программу партнёру с инструкцией, фиксирует замечания и повторяет тесты.

Продукт работы: Исходный код приложения, тестовый CSV, инструкция и протокол минимум восьми проверок.

Контроль: Основные команды работают, недопустимый ввод не повреждает данные, результаты воспроизводимы.

4.8. Защита программы и повторная проверка

Объём: 4 ч. Теория: 0 ч. Практика: 4 ч.

Теоретическое содержание

Защита показывает связь условия, алгоритма, кода и теста. Небольшое изменение правила подтверждает понимание автора. Демонстрация выполняется на учебных данных в локальной среде. Известное ограничение указывается прямо; отсутствие проверки не позволяет утверждать, что программа работает для любых данных.

Практическая работа

Участники запускают проект из чистой рабочей папки, проверяют инструкцию и комплектность. На защите объясняют одну функцию, показывают обработку ошибочного ввода и выполняют небольшую смену порога или формата отчёта. При необходимости повторно подтверждают недостающее умение после обратной связи. Финальная версия сохраняется вместе с тестами и обезличенными учебными примерами.

Продукт работы: Комплект запускаемого проекта и личная демонстрация.

Контроль: Автор объясняет функцию, тест и обоснованное изменение программы.

5. Воспитание

Цель воспитательной работы: Развивать ответственность за корректность кода и сохранность данных, честное представление результатов и уважительное техническое общение.

• Обсуждение последствий неверного ввода на безопасных вымышленных примерах.

• Рецензирование с конкретным тестом вместо личной оценки автора.

• Указание источников и сохранение истории версий работы.

Период	**Мероприятие и форма работы**	**Ожидаемое проявление**
1-2 недели	Договор учебной папки и понятного кода.	Участник не запускает неизвестные команды и сохраняет свою версию.
7 неделя	Проверка подсказок пользователю.	Сообщения становятся понятными другому человеку.
17 неделя	Обсуждение пустого набора без сокрытия ошибки.	Участник честно показывает непройденный тест и исправляет причину.
21-24 недели	Сохранение исходных учебных данных.	Отчёт создаётся отдельно, оригинал проверяется после операции.
27 неделя	Парная рецензия функций.	Замечание воспроизводимо, автор отвечает предметно.
33-34 недели	Защита с указанием ограничений.	Участник различает подтверждённый результат и будущие планы.

Обсуждаются действия: сохранение файла, честная запись теста, уважительная помощь. Личностные качества и убеждения не оцениваются баллами. Педагог использует наблюдения для улучшения организации работы, не формирует рейтинг «способностей к программированию».

6. Планируемые результаты

6.1. Предметные результаты

• Запускает локальные .ру-файлы и читает основные сообщения об ошибке.

• Выбирает тип, выполняет преобразование и выводит результат с понятной подписью.

• Реализует ветвления и завершающиеся циклы с проверенными границами.

• Обрабатывает список словарей и сохраняет исходную коллекцию при фильтрации.

• Выделяет минимум три функции с параметрами и возвращаемыми значениями.

• Читает и сохраняет CSV с явной кодировкой и проверкой структуры.

• Обрабатывает ожидаемый неверный ввод и отсутствующий файл без потери данных.

• Создаёт минимум восемь тестов и объясняет подтверждённое исправление.

6.2. Метапредметные результаты

• Трассирует состояние и отделяет причину от проявления ошибки.

• Планирует интерфейсы функций и структуру данных.

• Обосновывает замечание конкретным тестом.

• Документирует запуск, допустимые входы и ограничения.

6.3. Личностные результаты

• Не запускает непонятный код и не использует реальные личные данные в учебной работе.

• Сохраняет авторство и уважительно обсуждает решения.

• Честно отражает неудачные проверки и возвращается к исправлению.

Предметные результаты подтверждаются рабочими продуктами, наблюдением и оценочными заданиями из раздела 9. Личностные результаты обсуждаются в самооценке и педагогическом наблюдении без отметок за убеждения, сравнения личности ребёнка с другими детьми и публичного рейтинга.

7. Ресурсное обеспечение

7.1. Кадровые и материальные условия

Педагог, владеющий базовым Python, стандартной библиотекой csv, функциями, обработкой ошибок и начальным тестированием; подготовлен к безопасной организации компьютерных занятий.

Компьютерный кабинет с местом безэкранной работы и доступным локальным хранилищем учебных файлов.

• 12-15 компьютеров желательно; минимум 8 при организованной ротации и индивидуальном контроле.

• Оборудование, совместимое с выбранной поддерживаемой версией Python 3 и средой Thonny либо IDLE; сложная графика и выделенная видеокарта не нужны.

• Компьютер и экран педагога, локальные папки с резервным копированием, карточки тестов и учебные CSV.

7.2. Программное обеспечение и доступные варианты

• Поддерживаемая версия Python 3 с официального python.org; перед годом педагог проверяет совместимость среды и оборудования.

• Thonny с официального thonny.org либо входящая в поставку Python среда IDLE.

• Стандартная библиотека Python, включая csv; обязательные внешние пакеты, платные сервисы и детские аккаунты отсутствуют.

При отсутствии интернета используются заранее установленные интерпретатор, локальная справка и учебные файлы.

Если Thonny недоступен, применяется IDLE или установленный текстовый редактор с локальным запуском; содержание языка сохраняется.

При ограниченном оснащении планирование и рецензия выполняются без экрана, но каждый лично пишет и запускает контрольный фрагмент.

7.3. Методическое обеспечение

Чтение и прогноз предшествуют набору кода. Педагог предлагает короткий фрагмент, ученики заполняют таблицу состояний, затем проверяют интерпретатором. После объяснения они изменяют условие и создают собственную задачу. Такой порядок сокращает механическое копирование и делает видимым понимание типов, границ и порядка действий.

Новые конструкции вводятся через ограниченную задачу. На этапе input допускается оговорённый корректный ввод, а обработка ошибок добавляется отдельным модулем после освоения условий и функций. Педагог не маскирует незнакомыми библиотеками недостающие знания. Стандартная библиотека выбирается по назначению, внешние пакеты для обязательных работ не требуются.

Тестирование начинается с заранее вычисленного результата. Ученик записывает вход, ожидание и факт, включая пустой набор и граничное значение. После исправления выполняется весь прежний набор. assert используется в учебных тестах вычислительных функций; проверка пользовательских данных реализуется обычными условиями и обработкой конкретных исключений.

При рецензировании один участник запускает код другого на предоставленных учебных данных и формулирует воспроизводимый пример. Автор сам вносит изменение и объясняет его. Поддержка включает стартовый каркас, справку синтаксиса и готовый CSV, но итоговая защита требует личного анализа и небольшого изменения функции.

7.4. Безопасность

• Режим экранной работы и перерывов определяется организацией по возрасту и действующим требованиям; занятия включают безэкранную трассировку и обсуждение.

• Программы выполняются только в выделенной учебной папке; удаление системных файлов, массовые операции и изменение настроек компьютера не изучаются.

• Не запускается код из неизвестных источников; назначение каждой используемой команды должно быть понятно педагогу и ученику.

• Файловые операции ограничены предоставленными учебными данными; исходники сохраняются отдельно от создаваемых отчётов.

• Реальные пароли, токены, контакты, медицинские и финансовые данные не вводятся; все записи вымышленные и неперсональные.

• Сеть, боты, рассылки, сбор данных сайтов и внешние API в обязательный курс не входят; аккаунты несовершеннолетних не требуются.

• Установка среды и обновлений выполняется организацией из официальных источников; ученики не принимают неожиданные разрешения и не отключают защиту.

Перед каждым новым видом практики педагог проводит целевой инструктаж, проверяет исправность оборудования и понятность сигнала остановки. При неисправности или небезопасных условиях работа прекращается; обучающийся обращается к педагогу. Сведения об инструктажах оформляются по локальному порядку.

7.5. Индивидуализация и специальные условия

Предоставляются крупный шрифт, контрастная тема, памятки синтаксиса и готовые таблицы трассировки. При трудностях печати используется короткий каркас с пропущенными осмысленными фрагментами. Результат проверяется объяснением и изменением алгоритма, а не числом набранных строк или скоростью клавиатуры.

Дополнительное время выделяется на чтение сообщения об ошибке, сохранение и тесты. Инструкции дробятся, контрольные данные уменьшаются без исключения основных случаев. Парная помощь допустима для ввода, но выбор условия, функции и проверки остаётся за участником. Педагог отдельно фиксирует помощь и достигнутое понимание.

Особые условия обсуждаются с ребёнком и семьёй, при необходимости со специалистами организации. Диагноз не требуется для обычной педагогической поддержки. При ограничении экранного времени используются бумажные трассировки и устные разборы с короткими личными запусками; форма подтверждения результата согласуется заранее и сохраняет предметный смысл.

При обучении ребёнка с ОВЗ или инвалидностью условия и, при необходимости, адаптированная программа разрабатываются индивидуально с учётом рекомендаций ПМПК и ИПРА, если они применимы, доступности среды и возможностей организации. Медицинские сведения не обсуждаются публично. Участие в программе не подменяет необходимую коррекционную помощь.

8. Список литературы и электронных ресурсов

8.1. Нормативные и методические источники

Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации», статьи 2, 12, 28, 30 и 75. Применяется в действующей редакции.

http://publication.pravo.gov.ru/Document/View/0001201212300007

Приказ Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

http://publication.pravo.gov.ru/Document/View/0001202209270013

Методические рекомендации по разработке дополнительных общеразвивающих программ, в том числе в части интеграции с учебными предметами «Труд (технология)», «Музыка», «Изобразительное искусство», «Физическая культура». Письмо Минпросвещения России от 23.01.2026 № АБ-254/06. Рекомендуемая основа структуры и содержания.

https://gal.edummr.ru/wp-content/uploads/2026/02/metodicheskie_rekomendatsii_254_06_23_01_2026.pdf

СП 2.4.2.4283-26, утверждённые постановлением Главного государственного санитарного врача РФ от 02.06.2026 № 19. Действуют с 01.09.2026.

http://publication.pravo.gov.ru/document/0001202606020083

СанПиН 1.2.3685-21, постановление Главного государственного санитарного врача РФ от 28.01.2021 № 2. Установленный срок действия до 01.03.2027; применимые нормы на последующий период необходимо уточнить до этой даты.

http://publication.pravo.gov.ru/Document/View/0001202102030022

8.2. Ресурсы по содержанию программы

Python Documentation. The Python Tutorial. Официальная справка педагога по языку; руководство предполагает некоторый опыт, поэтому ученикам даются адаптированные короткие задания.

https://docs.python.org/3/tutorial/

Python Documentation. More Control Flow Tools. Проверка семантики range, ветвлений, циклов и функций.

https://docs.python.org/3/tutorial/controlflow.html

Python Documentation. Errors and Exceptions. Проверка различия синтаксических ошибок, исключений и их конкретной обработки.

https://docs.python.org/3/tutorial/errors.html

Python Documentation. csv. Официальное описание чтения и записи CSV, используемое при подготовке файловых заданий.

https://docs.python.org/3/library/csv.html

Thonny. Python IDE for beginners. Официальный источник среды, проверка установки и возможностей пошагового выполнения.

https://thonny.org/

Педагог заранее проверяет доступность выбранного ресурса, условия использования и возрастные ограничения. При изменении сервиса сохраняет образовательный результат через разрешённый офлайн-материал или доступное средство из раздела 7.

9. Оценочные материалы

9.1. Входная диагностика

Короткие бумажные и компьютерные пробы первых занятий уточняют стартовые навыки без требования знать Python.

№	Вопрос или задание	Ключ или ожидаемое действие	Балл
1	Есть 17 деталей, в комплекте 5. Сколько полных комплектов и какой остаток?	3 полных комплекта, остаток 2.	1
2	Сначала значение равно 2, затем его увеличили на 3 и удвоили. Каков результат?	10.	1
3	Условие «значение не меньше десяти» выполняется для десяти?	Да, граница включена.	1
4	Какие данные нужны программе расчёта площади прямоугольника?	Длина и ширина; результат равен их произведению.	1
5	Где безопасно сохранить учебный файл?	В своей согласованной рабочей папке под понятным именем, не заменяя чужую работу.	1
6	Почему одного удачного запуска недостаточно?	Другие входы, границы и ошибки могут вызвать иное поведение, нужны дополнительные тесты.	1

Максимум 6. 0-2: больше трассировки и опорных алгоритмов; 3-4: справка синтаксиса и промежуточная проверка; 5-6: самостоятельный старт. Результат влияет на поддержку, не на право продолжать курс.

9.2. Промежуточная работа

Период: к 17-й учебной неделе.

Написать анализатор списка [12, 9, 15, 9, 20]. Вывести количество, сумму, среднее, минимум, максимум и количество значений не ниже 12. Для пустого списка вывести сообщение «Нет данных», не делить на ноль и не искать минимум. Дополнительно проверить [0].

• Определить ожидаемые результаты вручную.

• Реализовать обход, накопление и проверку пустого набора.

• Выполнить три теста и сравнить факт с ожиданием.

• Объяснить один цикл и исправить найденное расхождение.

Критерий	Балл	Правила начисления баллов
Статистика	2	0: большинство результатов неверно; 1: часть верна; 2: для основного набора получены 5, 65, 13, 9, 20 и 3.
Граничные случаи	2	0: программа падает на пустоте; 1: пустота обработана, [0] неверен; 2: пустота даёт сообщение, [0] даёт количество 1, сумму 0, среднее 0, минимум 0, максимум 0, отбор 0.
Объяснение и тесты	2	0: личного объяснения нет; 1: объяснение с подсказкой или неполные тесты; 2: три теста записаны и цикл объяснён самостоятельно.

Максимум 6, достаточно 4 при личном объяснении. Недостающий случай исправляется и повторно проверяется при выделении функции в следующем занятии.

9.3. Итоговая работа и защита

Разработать локальное консольное приложение «Лаборатория измерений». Запись содержит непустое название и число от 0 до 1000; данные вымышленные. Команды: добавить, показать, вычислить количество/сумму/среднее/минимум/максимум, отобрать значения не ниже порога, сохранить и загрузить CSV, выйти. Минимум три функции с возвращаемым результатом. Базовые значения 12, 9, 15, 4 дают сумму 40, среднее 10, минимум 4, максимум 15; при пороге 10 выбираются 12 и 15.

Материалы к защите

• Исходные .py-файлы с понятными именами функций.

• Учебный CSV и отдельный файл тестов или подробный протокол.

• Инструкция запуска, команд и допустимого ввода.

• Журнал одного исправления с повторной проверкой.

Порядок выполнения

• Описать структуру записи, меню и правила ошибок.

• Спланировать функции ввода, обработки и сохранения.

• Реализовать минимальную версию на базовом наборе.

• Добавить проверки диапазона, пустоты и ожидаемых исключений.

• Проверить минимум восемь случаев: база, пустота, ноль, граница 1000, выход за диапазон, текст вместо числа, отсутствующий файл, сохранение с повторной загрузкой.

• Исправить сбой, повторить тесты и защитить личное решение.

Вопросы для защиты

• Почему input требует преобразования для числового расчёта?

• Чем return в твоей функции отличается от print?

• Как обрабатывается пустой список и почему это важно?

• Какой тест подтверждает, что сохранение не потеряло записи?

Критерий	**Балл**	**Уровни выполнения**
Функциональность	3	0: приложение не запускается; 1: работают отдельные команды; 2: основной сценарий работает с небольшими ограничениями; 3: все обязательные команды подтверждены тестами.
Структура кода	3	0: логика не объясняется; 1: неразделённый код с повторениями; 2: три функции с частичной зависимостью от внешнего состояния; 3: понятные функции, параметры и возвращаемые результаты.
Надёжность и тесты	3	0: проверки отсутствуют; 1: проверены 1-4 случая; 2: проверены 5-7 случаев либо восемь случаев с оставшимся сбоем; 3: все обязательные случаи обработаны и повторно проверены.

Критерий	Балл	Уровни выполнения
Документация и защита	3	0: личного объяснения нет; 1: только демонстрация; 2: инструкция и объяснение с подсказкой; 3: комплект запускается по инструкции, автор самостоятельно меняет и проверяет фрагмент.

Максимум 12, достаточно 8 при личном объяснении, сохранности учебных исходников и безопасном составе данных. Повторная проверка недостающего результата проводится на неделе 34 после обратной связи.

Педагог фиксирует умение, тест, результат, помощь и следующий шаг. Код и данные сохраняются в учебной папке без реальных личных сведений; публичный рейтинг участников не составляется.

10. Методические материалы

10.1. Анализ списка без падения на пустоте

Цель: Проверить статистику для обычного, пустого и минимального набора.

Материалы: Python, редактор, наборы [12,9,15,9,20], [] и [0].

Порядок работы

1. Вручную записать ожидаемую сумму 65 и среднее 13 для первого набора.
2. До вычислений проверить, есть ли элементы.
3. Для непустого списка определить количество, сумму, минимум и максимум.
4. Разделить сумму на количество только в непустой ветви.
5. Проверить отбор значений не ниже 12 с включённой границей.
6. Запустить все три набора и объяснить каждый результат.

Проверка: Основной набор даёт 5,65,13,9,20,3; пустой даёт сообщение; [0] обрабатывается как одно значение.

Дополнительная задача: Выделить функцию, возвращающую словарь статистики или специальный результат для пустоты.

Методический комментарий: Проверка «если сумма равна нулю» не заменяет проверку пустоты: список [0] содержит данные.

10.2. Безопасный круг CSV

Цель: Сохранить и восстановить учебные записи без потери структуры.

Материалы: Модуль csv, локальный файл name,value с вымышленными названиями.

Порядок работы

1. Прочитать заголовок и проверить обязательные поля.
2. Преобразовать числовое значение и проверить диапазон.
3. При ошибке сообщить строку и не заменять текущий набор неполной загрузкой.
4. Сохранить результат под новым именем с UTF-8.
5. Прочитать сохранённый файл повторно.
6. Сопоставить количество, названия и значения с исходным набором.

Проверка: Число и содержимое записей совпадают; исходный файл не изменён.

Дополнительная задача: Добавить название с запятой и убедиться, что модуль csv корректно обрабатывает кавычки.

Методический комментарий: Ручное split по запятой не подходит для общего CSV с запятыми внутри полей; использовать штатный модуль.

10.3. Тест против ошибки на границе

Цель: Обнаружить неверный знак сравнения в фильтре.

Материалы: Функция отбора, набор [9,10,11], порог 10.

Порядок работы

1. Записать ожидаемый результат [10,11].
2. Выполнить функцию с условием больше порога и получить [11].
3. Назвать отличие требования «не ниже» от «выше».
4. Исправить сравнение на включающее равенство.
5. Повторить тест, затем проверить пустой список и повторяющиеся десятки.

Проверка: Граница включена, пустой набор даёт пустой результат, повторы сохраняются.

Дополнительная задача: Добавить отдельный тест порога ниже всех значений.

Методический комментарий: Не заменять тест только данными, удалёнными от границы: такой набор не обнаружит исходную ошибку.

10.4. Основные понятия

Понятие	**Пояснение**
Интерпретатор	Программа, выполняющая код языка Python.
Тип	Категория значения, определяющая допустимые операции.
Присваивание	Связывание имени с вычисленным значением.
Ветвление	Выбор части алгоритма по логическому условию.
Итерация	Одно выполнение тела цикла.
Список	Упорядоченная изменяемая коллекция элементов.
Словарь	Коллекция связей ключей и значений.
Функция	Именованный фрагмент программы с параметрами и возможным результатом.
Исключение	Сигнал об особой ситуации, возникшей при выполнении.
Граничный тест	Проверка значения на границе допустимого диапазона или рядом с ней.
Регрессионная проверка	Повтор прежних тестов после изменения программы.