

**«Ақмола облысы білім басқармасының Сандықтау ауданы бойынша білім бөлімі Максимовка ауылының орта білім беретін мектебі» КММ
КГУ «Общеобразовательная школа села Максимовка отдела образования по Сандыктаускому району управления образования Акмолинской области»**

**Андреева Ольга Филипповна
учитель информатики**

Авторская программа

«Обучение программированию на Python детей с особыми образовательными потребностями в 6 классе»

2025 г.

Пояснительная записка

Данная программа ориентирована на развитие у детей с особыми образовательными потребностями (ООП) навыков логического мышления, алгоритмики, творческого подхода и базового программирования на языке Python. Она сочетает элементы дополнительного образования в сфере информационных технологий с учетом психолого-педагогических особенностей обучающихся с ООП.

Программа имеет социально-педагогическую направленность и способствует формированию компетенций цифровой грамотности, коммуникации и самореализации в доступной и поддерживающей среде.

Новизна заключается в интеграции адаптированных методик обучения программированию с учетом когнитивных, речевых, сенсорных и поведенческих особенностей детей с ООП. Использование Python как простого, интуитивно понятного языка с мощными возможностями позволяет вовлекать детей в реальное программирование даже на начальном уровне.

Актуальность обусловлена:

- растущей потребностью в цифровых навыках у всех категорий населения;
- необходимостью инклюзивного образования в ИТ-сфере;
- недостаточным количеством адаптированных программ по программированию для детей с ООП.

Педагогическая целесообразность программы подтверждается возможностью индивидуализации темпа и формы подачи материала, включением визуальных и игровых компонентов, а также широким спектром применимости навыков программирования как в образовательной, так и в социальной адаптации обучающихся.

Цели и задачи программы:

Цель: сформировать у детей с особыми образовательными потребностями базовые знания и умения в области программирования на языке Python, способствовать развитию логического и алгоритмического мышления, коммуникационных и метапредметных навыков.

Задачи:

- ознакомить с основами языка Python (переменные, операторы, типы данных);
- научить писать простые программы и решать базовые задачи;
- развивать навыки логического анализа, работы с алгоритмами;
- формировать уверенность в собственных возможностях, усиливать мотивацию к обучению;
- обеспечить доступную и комфортную учебную среду с опорой на наглядность, поэтапность и поддержку.

Особенности реализации программы:

Адаптированная подача материала: использовать поэтапные инструкции, нахождение ошибок, образцы, игровые моменты.

Индивидуализация и гибкость: темп и сложность заданий регулируются под возможности каждого обучающегося.

Использование интерактивных сред: среда программирования с подсветкой синтаксиса и автозавершением (например, Thonny или online-платформы).

Психолого-педагогическое сопровождение: тесное взаимодействие педагога с родителями и специалистами сопровождения (психолог, дефектолог, логопед).

Ожидаемые результаты и способы их проверки

Ожидаемые результаты:

- обучающиеся знают базовые конструкции языка Python и могут применять их на практике;
- способны решать простые задачи с использованием типов данных;
- развивают уверенность в себе и интерес к технологиям;
- проявляют навыки работы с алгоритмами и проектной деятельностью.

Формы и способы проверки:

- текущий контроль через интерактивные задания и мини-проекты;
- итоговое тестирование (в доступной форме);
- демонстрация собственных проектов или фрагментов кода;
- самооценка и рефлексия через анкетирование и обсуждения;
- наблюдение за прогрессом со стороны педагога и сопровождения.

Учебно-тематический план для 6 класса по авторской программе «Обучение программированию на Python детей с особыми образовательными потребностями» (в рамках 10 часов). План включает разбиение на теоретические и практические занятия, с учетом возрастных и когнитивных особенностей обучающихся.

План включает разбиение на теоретические и практические занятия, с учетом возрастных и когнитивных особенностей обучающихся.

Раздел: Программирование на языке Python (пайтон)
Учебно–тематический план

6 класс

№ п/п	Раздел/ Сквозные темы	Темы урока	Цели обучения	Количество часов теория	Количество часов практика
III четверть					
1-2	Программирование на языке Python (пайтон)	Знакомство с IDE (Integrated Development Environment (интегрированный средой разработки))	6.1.2.2 использовать возможности интерфейса IDE (Integrated Development Environment (интегрированный средой разработки))	1	1
3		Алфавит языка. Синтаксис	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)	0.5	0.5
4		Типы данных	6.3.3.1 классифицировать типы данных	0.5	0.5
5		Правила записи арифметических выражений	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)	0.5	0.5
6		Ввод и вывод чисел	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)	0.5	0.5
7		Программирование линейных алгоритмов	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)	0.5	0.5
8		Мини-проект: написание простой программы	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)	0.5	0.5
9-10		Защита проекта, рефлексия, повторение	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)	1	1
Итого:				5	5

Примечание: Программа рассчитана на обучающихся с ООП, поэтому соотношение теории и практики сбалансировано в пользу практических заданий, но с регулярным повторением и визуальной поддержкой. На каждом занятии предполагается работа с адаптированными средами (например, Thonny), а также использование цветных карточек, шаблонов кода и визуальных подсказок. Темы можно варьировать в зависимости от индивидуальных образовательных маршрутов и рекомендаций ПМПК.

Описание учебно тематического плана

Учебно-тематический план разработан в рамках авторской адаптированной программы дополнительного образования и предназначен для обучающихся 6 класса с особыми образовательными потребностями (ООП). План включает 10 учебных часов и реализуется в форме теоретико-практических занятий с применением современных методов и средств обучения. Программа направлена на формирование у обучающихся базовых представлений о программировании на языке Python, развитие логического и алгоритмического мышления, освоение простейших конструкций языка (переменные, типы данных, ввод/вывод, арифметические выражения, линейные алгоритмы). Особое внимание уделяется формированию устойчивой учебной мотивации, доступности материала и наглядности.

В программе предусмотрены:

- ввод в работу с **интегрированной средой разработки (IDE)**, освоение интерфейса;
- пошаговое знакомство с синтаксисом языка Python;
- освоение типов данных (строки, числа, логические значения);
- решение задач с линейной логикой — базовые арифметические и логические операции;
- ввод/вывод данных с клавиатуры и на экран;
- выполнение небольшого мини-проекта, который завершает курс.

Для закрепления материала используются:

- интерактивные упражнения;
- визуальные алгоритмы;
- парная и индивидуальная работа;
- задания с опорой на повседневные жизненные ситуации.

Форма итогового контроля — мини-проект, защита и обсуждение которого позволяют выявить уровень понимания, сформированность ключевых умений, а также уровень самостоятельности обучающегося в решении поставленных задач.

Курс не требует предварительной подготовки и позволяет вовлечь детей с различными вариантами ООП в начальное программирование через практико-ориентированную деятельность.

Описание тем:

**Тема 1-2 : Знакомство с IDE (интегрированная среда разработки)
(для детей с ООП)**

Цель обучения: использовать возможности интерфейса IDE (Integrated Development Environment (интегрированная среда разработки))

Теоретическая часть:

1. Что такое IDE:

Интегрированная среда разработки (IDE) — это программа, предназначенная для удобного написания, запуска и отладки кода.

IDE объединяет в себе редактор кода, консоль, средства запуска и другие инструменты, которые помогают программисту.

2. Обзор интерфейса:

- Главное меню;
- Окно редактирования кода;
- Консоль вывода;
- Кнопка «Запуск» (Run).

3. Основы безопасной работы за компьютером:

- Удобная поза;
- Перерывы каждые 15-20 минут;
- Безопасное обращение с техникой.

Практическая часть

Задача: Пошаговое решение и результат

Шаг 1: Открываем Thonny

Шаг 2: Вводим первую строку кода

В окне редактора введите следующий код:

```
python  
print("Привет, мир!")
```

Шаг 3: Нажимаем кнопку «Запуск»

Нажмите на зелёную кнопку ► «Запуск» в верхней части окна, либо нажмите **F5** на клавиатуре. Внизу, в консоли, должно появиться:

Привет, мир!

Шаг 4: Изменяем текст в print() на ваше имя

Например, если вас зовут Анна, замените строку на:

```
python
```

```
print("Привет, Анна!")
```

Нажмите снова «Запуск», и вы увидите:

Привет, Анна!

Совет: Вы можете заменить имя на любое другое, например:

```
python  
print("Привет, Максим!")
```

Индивидуальное практическое задание

Что нужно сдать:

- Программа с кодом `print("Привет, мир!")` и результатом его выполнения.
- Программа с изменённым кодом, где в `print()` указано ваше имя, и результат выполнения.

Форма проверки: демонстрация работы учителю, взаимопроверка, проверка спикером

Таблица критериев оценки (10 баллов)

Критерий	Баллы	Описание
Корректность написания кода (<code>print(...)</code>)	2	Использован правильный синтаксис, отсутствуют ошибки.
Использование IDE и запуск программы	2	Учащийся запустил программу через интерфейс среды разработки.
Правка строки с личным именем	2	В строке <code>print()</code> корректно указано собственное имя.
Наличие выполненной программы с результатами	2	Программа: с оригинальной строкой и с изменённой.
Соблюдение инструкции	2	Работа выполнена по инструкции
Максимум баллов	10	

Тема 3 : Алфавит языка. Синтаксис (для детей с ООП)

Цель обучения: записывать линейные алгоритмы на языке Python (пайтон)

Теоретическая часть:

1. Базовые элементы Python:

- Язык Python использует буквы (латинский алфавит), цифры, знаки препинания и спецсимволы (=, +, #, ", ', :, и др.).
- Имена переменных чувствительны к регистру (Возраст ≠ возраст).
- Примеры допустимых символов: a, z, A, Z, 0–9, _.

2. Правила именования переменных:

- Должны начинаться с буквы или знака подчеркивания _.
- Нельзя начинать с цифры.
- Нельзя использовать пробелы.
- Нельзя использовать ключевые слова Python (например: if, for, print).
- Имена должны быть понятными и описательными, например: имя, возраст_пользователя.

3. Комментарии и отступы:

- Комментарии начинаются с символа # и не исполняются программой.
- Отступы (обычно 4 пробела) важны для определения структуры кода (например, внутри циклов или условий).

Практическая часть

Шаг 1: Создание переменных

В среде разработки (например, Thonny) создайте три переменные:

```
python
```

```
имя = "Миша"  
возраст = 12  
любимый_цвет = "синий"
```

Шаг 2: Вывод переменных с пояснением

Напечатайте каждую переменную с поясняющим текстом

```
python
```

```
print("Имя ученика:", имя)  
print("Возраст:", возраст)  
print("Любимый цвет:", любимый_цвет)
```

Шаг 3: Создание переменной с ошибкой

Пример 1 (ошибка из-за пробела в имени переменной):

```
python  
мой возраст = 12
```

Ошибка IDE:

```
javascript  
SyntaxError: invalid syntax
```

Исправление:

```
python  
  
мой_возраст = 12
```

Пример 2 (ошибка: имя переменной не может начинаться с цифры):

```
python  
  
1имя = "Катя"
```

Ошибка IDE:

```
javascript  
  
SyntaxError: invalid syntax
```

Исправление:

```
python  
имя1 = "Катя"
```

Итоговый исправленный код:

```
python  
  
имя = "Миша"  
возраст = 12  
любимый_цвет = "синий"  
  
print("Имя ученика:", имя)  
print("Возраст:", возраст)  
print("Любимый цвет:", любимый_цвет)  
  
мой_возраст = 12  
имя1 = "Катя"
```

Индивидуальное практическое задание

Выполнить:

- Записать программу (переменные, вывод, ошибка и её исправление).
- Описать ошибку.
- Краткий комментарий (в комментариях к коду или отдельным текстом):
«Я допустил ошибку в названии переменной. IDE подсказала, что...»

Таблица критериев оценки (10 баллов)

Критерий	Баллы	Описание
Созданы три переменные с корректными именами	2	Имена переменных осмысленные, без синтаксических ошибок
Переменные выведены на экран с пояснениями	2	Используется print() с понятным текстом перед переменными
Попытка создания переменной с ошибкой	1	Создана хотя бы одна переменная с ошибочным именем
Анализ ошибки по сообщению IDE	2	Учащийся зафиксировал сообщение об ошибке и (в идеале) понял причину
Комментарии в коде (#) или пояснения к работе	1	Присутствуют комментарии к коду или отдельное текстовое объяснение
Оформление кода и структура	1	Аккуратный код: соблюдены отступы, нет лишнего
Наличие программ по заданиям	1	Работа продемонстрирована
Максимум баллов	10	

Тема 4 : Типы данных. (для детей с ООП)

Цель обучения: классифицировать типы данных

Представь:

Переменные — это как коробки, а **тип данных** — это то, что лежит внутри коробки:

- **строка (str)** — это текст, в кавычках, например: "Привет"
- **число (int или float)** — это целое число (10) или дробное (3.14)
- **логика (bool)** — это правда или ложь: True (да), False (нет)

Теория

Основные типы данных:

Тип данных	Как выглядит	Пример	Как узнать тип?
str	строка в кавычках	"Привет"	print(type(переменная))
int	целое число	10	
float	дробное число	3.14	
bool	логический тип	True, False	

Преобразование типов:

Что делаем	Код	Что получится
Строку в число	int("5")	5
Число в строку	str(7)	"7"

Сравнение:

Можно сравнивать значения. Пример:

```
python  
print(5 > 3) # будет True, потому что 5 больше 3
```

Практика: пошагово

Открой IDE (например, Thonny) и выполни следующие задания:

Задание 1: Создай переменные разных типов

```
python  
текст = "Привет"  
число = 10  
правда = True
```

Задание 2: Выведи тип каждой переменной

```
python
текст = "Привет, мир!"
число = 42
правда = True
```

```
print(type(текст))
print(type(число))
print(type(правда))
```

Вывод на консоли:

```
<class 'str'>
<class 'int'>
<class 'bool'>
```

Задание 3: Преобразуй один тип данных в другой

```
python
строка_число = "5"
число_из_строки = int(строка_число)

число = 7
строка_из_числа = str(число)
```

```
print(число_из_строки)
print(строка_из_числа)
```

Вывод на консоли:

```
5
7
```

Задание 4: Сравни два значения

```
python

print(5 > 3) # True
print(10 == 7) # False
```

Что делает код:

- $5 > 3$ → проверяет, больше ли 5, чем 3 → **результат: True**
- $10 == 7$ → проверяет, равны ли 10 и 7 → **результат: False**

Вывод в консоли:

```
True
False
```

Дополнительно: Добавь комментарии к каждой строке с помощью #, чтобы пояснить, что ты делаешь.

Результат урока:

- Программа с кодом.
- Комментарии в коде (если не получается — можно устно на уроке).

Таблица критериев оценки (10 баллов)

Критерий	Баллы	Комментарий
Созданы переменные трёх типов (str, int, bool)	2	Переменные названы понятно и корректно
Использована функция type() для проверки типов	2	Выводит тип каждой переменной
Преобразование типов выполнено верно	2	Есть int("5") и str(7)
Сравнение значений работает	2	Использовано сравнение (>, ==) и выведен результат
Комментарии в коде или пояснение на уроке	1	Простые пояснения — что делает каждая строка
Опрятный код и сдача в срок	1	Нет лишнего, код читаемый, всё сдано вовремя
Максимум баллов	10	

Тема 5 : Правила записи арифметических выражений (для детей с ООП)

Цель обучения: записывать линейные алгоритмы на языке Python (пайтон)

Теория:

- Арифметические операторы: +, -, *, /, //, %, **.
- Приоритет операций.

Практика:

- Составление выражений и расчет результатов.
- Использование переменных в формулах.
- Создание "калькулятора".

Практическое задание

- **Цель:** освоить арифметику в Python.
Задания: Посчитай: $5 + 7$, $12 - 4$, $3 * 3$, $10 / 2$.
- Используй переменные:
В программе python
яблоки = 5
друзей = 2
print(яблоки / друзей)
- Создай «секретный пример»: придумай арифметику для друга и выведи ответ.

Выполнение задания:

1. Простые арифметические выражения:

```
python
```

```
print(5 + 7) # 12  
print(12 - 4) # 8  
print(3 * 3) # 9  
print(10 / 2) # 5.0
```

2. Использование переменных:

```
python
```

```
яблоки = 5  
друзей = 2  
print(яблоки / друзей) # 2.
```

3. «Секретный пример» (авторская задача):

Свой пример.

```
python
монеты = 20
карманы = 4
print("В каждом кармане по", монеты / карманы, "монет") # 5.0
```

Весь код вместе:

```
python
```

- **Арифметика**

```
print(5 + 7)
print(12 - 4)
print(3 * 3)
print(10 / 2)
```

- **Переменные**

```
яблоки = 5
друзей = 2
print(яблоки / друзей)
```

- **Секретный пример**

```
монеты = 20
карманы = 4
print("В каждом кармане по", монеты / карманы, "монет")
```

Таблица критериев оценки (10 баллов)

Критерий	Баллы	Комментарий
Созданы переменные трёх типов (str, int, bool)	2	Переменные названы понятно и корректно
Использована функция type() для проверки типов	2	Выводит тип каждой переменной
Преобразование типов выполнено верно	2	Есть int("5") и str(7)
Сравнение значений работает	2	Использовано сравнение (>, ==) и выведен результат
Комментарии в коде или пояснение на уроке	1	Простые пояснения — что делает каждая строка
Опрятный код и сдача в срок	1	Нет лишнего, код читаемый, всё сдано вовремя
Максимум баллов	10	

Тема 6 : Ввод и вывод чисел в Python (для детей с ООП)

Подходит для детей 9–13 лет, в том числе с задержкой развития, нарушениями речи или внимания.

Простые слова, крупный текст, пошаговые действия, доброжелательные примеры.

Цель обучения: записывать линейные алгоритмы на языке Python (пайтон)

✓ Что такое Python?

- **Python** — это язык, на котором можно писать программы. Он похож на Lego — ты строишь команды и они **работают!**

✓ Что такое input()?

- Это команда, которая **спрашивает** у пользователя что-то.

```
python
```

```
имя = input("Как тебя зовут? ")
```

Пользователь вводит: Маша

- Программа запомнила: имя = "Маша"
- ✓ **Что такое print()?**
- Это команда, которая **пишет на экран.**

```
python
```

```
print("Привет, ", имя)
```

Вывод: Привет, Маша

Работа с числами

Важно: Когда мы получаем число через input, оно всегда **строка**. Нужно перевести в **число**:

```
python
```

```
возраст = int(input("Сколько тебе лет? "))
```

Пошаговая программа №1: Сколько тебе лет?

```
python
```

Шаг 1. Спрашиваем возраст

```
возраст = int(input("Сколько тебе лет? "))
```

Шаг 2. Прибавляем 1 год

```
новый_возраст = возраст + 1
```

Шаг 3. Показываем результат

```
print("Через год тебе будет", новый_возраст)
```

Объяснение:

Ребёнок вводит, например, 10 → программа считает $10 + 1 = 11$ → показывает.

Пошаговая программа №2: Сколько яблок осталось?

```
python
```

Шаг 1. Сколько у тебя яблок?

```
яблоки = int(input("Сколько у тебя яблок? "))
```

Шаг 2. Сколько ты отдал другу?

```
отдал = int(input("Сколько ты отдал другу? "))
```

Шаг 3. Считаем остаток

```
осталось = яблоки - отдал
```

Шаг 4. Показываем результат

```
print("У тебя осталось", осталось, "яблок")
```

Пример:

Ты ввёл: 8 яблок → отдал 3 → осталось 5

Пошаговая программа №3: Приветствие

```
python
```

Спрашиваем имя

```
имя = input("Как тебя зовут? ")
```

Приветствуем

```
print("Привет,", имя + "!")
```

Это делает программу живой и доброжелательной.

Программа-комбайн (всё вместе)

```
python
```

```
имя = input("Как тебя зовут? ")
```

```
print("Привет,", имя + "!")
```

```
возраст = int(input("Сколько тебе лет? "))
```

```
print("Через год тебе будет", возраст + 1)
```

```
яблоки = int(input("Сколько у тебя яблок? "))
```

```
отдал = int(input("Сколько ты отдал другу? "))
print("У тебя осталось", яблоки - отдал, "яблок")
```

Таблица критериев оценки (10 баллов)

Критерий	Баллы	Комментарий
Ввод данных (input)	1	Используется функция input() для получения данных от пользователя
Вывод данных (print)	1	Используется функция print() для отображения информации
Преобразование типов (int)	1	Применяется int() для перевода строки в число
Арифметические операции	1	Есть базовые вычисления (сложение, вычитание и т.д.)
Простота и доступность	1	Код понятный, легко читается и воспринимается
Пошаговое объяснение	1	Есть логическое построение и объяснение каждого этапа
Комментарии к коду	1	Присутствуют поясняющие комментарии
Гибкость (имя, возраст, счёт)	1	Работает с разными переменными, данными
Работа с текстом + числами	1	Используются и строки, и числа в программе
Интерактивность	1	Взаимодействие с пользователем реализовано

Максимум баллов 10

Карточки-команды для учащихся:

1. input()

Что делает: спрашивает у пользователя что-то

Пример:

```
python
имя = input("Как тебя зовут?")
```

Программа спрашивает: «Как тебя зовут?»

ученик вводит: Маша

Теперь имя = "Маша"

2. print()

Что делает: выводит на экран сообщение

Пример:

```
python
```

```
print("Привет,", имя)
```

Программа говорит: «Привет, Маша»

3. int()

Что делает: превращает строку в число

Пример:

```
python
```

```
возраст = int(input("Сколько тебе лет?"))
```

Если ввёл "10", программа превращает в число 10

Пошаговые программы

Программа №1: Сколько тебе лет?

```
python
```

```
возраст = int(input("Сколько тебе лет?"))
```

```
новый_возраст = возраст + 1
```

```
print("Через год тебе будет", новый_возраст)
```

Ввёл: 10

Прибавили 1

Ответ: «Через год тебе будет 11»

Программа №2: Сколько яблок осталось?

```
python
```

```
яблоки = int(input("Сколько у тебя яблок?"))
```

```
отдал = int(input("Сколько ты отдал другу?"))
```

```
осталось = яблоки - отдал
```

```
print("У тебя осталось", осталось, "яблок")
```

Ввёл: 8

Отдал: 3

Ответ: «У тебя осталось 5 яблок»

Программа №3: Приветствие

```
python
```

```
имя = input("Как тебя зовут?")
```

```
print("Привет,", имя + "!")
```

Ввела: Катя

Ответ: «Привет, Катя!»

Программа-комбайн № 4: (всё вместе)

```
python
```

```
имя = input("Как тебя зовут? ")
```

```
print("Привет,", имя + "!")
```

```
возраст = int(input("Сколько тебе лет? "))
```

```
print("Через год тебе будет", возраст + 1)
```

```
яблоки = int(input("Сколько у тебя яблок? "))
```

```
отдал = int(input("Сколько ты отдал другу? "))
```

```
print("У тебя осталось", яблоки - отдал, "яблок")
```

Методические рекомендации для педагога

Приём

Подсказка

Цветные карточки

Распечатайте print, input, int на цветной бумаге

Паузы и объяснения

После каждой строки делайте паузу и объясняйте, что она делает

Рисунки и эмодзи

Используйте визуальные схемы: $\square + \text{🍏} - \text{🍏} = \text{🍏}$

Индивидуальные примеры

Пусть дети вводят свои имена, любимые фрукты — это делает урок личным

Повторения

Повторите команды несколько раз в разных ситуациях (имя, возраст, яблоки)

Мини-диалоги

Ставьте код как диалог: программа спрашивает — ребёнок отвечает — программа считает

Тема 7 : Программирование линейных алгоритмов (без условий) (для детей с ООП)

Цель обучения: записывать линейные алгоритмы на языке Python (пайтон)

Теория: Что такое линейный алгоритм?

- **Линейный алгоритм** — это набор действий, которые выполняются друг за другом, шаг за шагом, без никаких "если" или "пока".
- Представь, что ты следуешь рецепту: сначала кладёшь муку, потом сахар, потом яйца — в точном порядке.

Последовательность действий очень важна!

Практика: Простые программы на Python

Программа 1: Сколько денег останется?

```
python
# Спросить сколько у тебя денег
деньги = int(input("Сколько у тебя денег? "))

# Спросить сколько потратил
потратил = int(input("Сколько ты потратил? "))

# Посчитать, сколько осталось
осталось = деньги - потратил

# Вывести результат на экран
print("Осталось:", осталось)
```

Что делает эта программа?

- Спрашивает у тебя два числа: сколько у тебя денег и сколько ты потратил.
- Вычисляет, сколько денег осталось (вычитает).
- Показывает ответ.

Программа 2: Площадь прямоугольника

```
python
# Ввод длины прямоугольника
длина = int(input("Введите длину прямоугольника (в см): "))

# Ввод ширины прямоугольника
ширина = int(input("Введите ширину прямоугольника (в см): "))
```

```
# Вычисление площади (длина умножить на ширину)
площадь = длина * ширина
# Показать площадь на экране
print("Площадь прямоугольника:", площадь, "кв.см")
```

Объяснение:

- Получаем два числа — длину и ширину.
- Умножаем их.
- Выводим результат.

Программа 3: Перевод сантиметров в метры

```
python
# Ввод длины в сантиметрах
см = int(input("Введите длину в сантиметрах: "))
# Переводим сантиметры в метры
метры = см / 100
# Показываем результат
print("Это", метры, "метров")
```

Что происходит?

- Вводим длину в сантиметрах.
- Делим на 100, чтобы получить метры.
- Показываем результат.

Таблица критериев оценки (10 баллов)

Критерий	Баллы	Комментарий
1. Использование переменных	2	Переменные названы понятно и используются правильно
2. Ввод данных через input()	2	Программа запрашивает данные у пользователя
3. Преобразование типов с int()	2	Корректное преобразование строк в числа
4. Арифметические операции	2	Выполнены правильные вычисления (+, -, *)
5. Вывод результата с print()	1	Результаты выводятся понятным сообщением
6. Комментарии в коде	1	Код сопровождается комментариями, понятными для детей
Итого	10	

Практическое задание для детей

1. Написать программу «Сколько денег останется», как в примере.
2. Написать программу «Площадь прямоугольника» с вводом длины и ширины.
3. Написать программу «Перевод см в метры».

Полные решения для каждого задания

Задание 1: «Сколько денег останется»

```
python
```

```
деньги = int(input("Сколько у тебя денег? "))  
потратил = int(input("Сколько ты потратил? "))  
осталось = деньги - потратил  
print("Осталось:", осталось)
```

Задание 2: «Площадь прямоугольника»

```
python
```

```
длина = int(input("Введите длину прямоугольника (в см): "))  
ширина = int(input("Введите ширину прямоугольника (в см): "))  
площадь = длина * ширина  
print("Площадь прямоугольника:", площадь, "кв.см")
```

Задание 3: «Перевод см в метры»

```
python
```

```
см = int(input("Введите длину в сантиметрах: "))  
метры = см / 100  
print("Это", метры, "метров")
```

Дополнительные рекомендации для педагога

- Используйте **яркие карточки с командами** (input, print, int).
- Объясняйте каждую строку кода простыми словами.
- Делайте паузы, чтобы дети успели понять и повторить.
- Поощряйте детей вводить свои числа — это повышает интерес.
- Используйте рисунки и эмодзи для визуализации (например, деньги , прямоугольник , сантиметры )

Тема 8: Мини-проект: написание простой программы (для детей с ООП)

Цель обучения: записывать линейные алгоритмы на языке Python (пайтон)

Теория

Повторяем, что уже знаем

- **Переменные** — это "коробочки", в которых хранятся данные (имя, число).
- **Ввод данных** — команда `input()`, чтобы спросить у пользователя что-то.
- **Вывод данных** — команда `print()`, чтобы показать информацию на экране.
- **Преобразование типов** — например, `int()` превращает слово с цифрами в настоящее число.
- **Арифметика** — мы умеем складывать, вычитать, умножать и делить числа.
- **Линейный алгоритм** — это когда программа выполняет команды одна за другой без "если" или циклов.

Планирование проекта

Перед тем как писать программу, нужно подумать:

Вопрос	Что нужно сделать
Цель проекта	Что программа должна делать?
Вводные данные	Какие данные нужно спросить у пользователя?
Логика программы	Какие действия надо выполнить с этими данными?
Вывод результата	Что показать пользователю?

Практика

Задание: Написать свою простую программу на выбор

Варианты мини-проектов

Проект	Что нужно сделать
Приветствие	Спросить имя и возраст, вывести приветствие
Магазин	Спросить цену и количество товара, посчитать и показать сумму
Калькулятор	Спросить два числа и действие (+, -, *, /), вывести ответ
Обмен валюты	Спросить сумму в тенге, перевести в доллары по курсу

Таблица критериев оценки (10 баллов)

Критерий	Баллы	Комментарий
1. Планирование программы	2	Цель, вводные данные и логика прописаны и понятны
2. Использование переменных	2	Переменные названы понятно и используются корректно
3. Ввод данных с input()	1	Пользователь может ввести данные
4. Преобразование типов (например, int())	1	Корректное преобразование строки в число
5. Арифметические операции	2	Выполнены правильные вычисления
6. Вывод данных с print()	1	Результат показан понятно
7. Комментарии в коде	1	В коде есть понятные комментарии
Итого	10	

Примеры программ с комментариями

1. Программа «Приветствие»

```
python
```

```
# Спросить имя пользователя  
имя = input("Как тебя зовут? ")
```

```
# Спросить возраст пользователя и преобразовать в число  
возраст = int(input("Сколько тебе лет? "))
```

```
# Вывести приветствие с именем и возрастом  
print("Привет,", имя + "!")  
print("Тебе уже", возраст, "лет.")
```

2. Программа «Магазин»

```
python
```

```
# Спросить цену товара  
цена = float(input("Введите цену товара: "))
```

```
# Спросить количество товара
количество = int(input("Введите количество товара: "))

# Посчитать общую стоимость
итого = цена * количество

# Показать итоговую сумму
print("Общая стоимость:", итого)
```

3. Программа «Обмен валюты»

```
python

# Спросить сумму в тенге
тенге = float(input("Введите сумму в тенге: "))

# Курс обмена (пример)
курс = 470.0 # 1 доллар = 470 тенге

# Перевести тенге в доллары
доллары = тенге / курс

# Вывести результат
print("Это примерно", round(доллары, 2), "долларов")
```

Методические рекомендации для педагога

- Используйте **цветные карточки с командами** (input, print, int, арифметика).
- Помогайте детям **планировать программу на бумаге** перед кодом.
- Объясняйте код по шагам, показывайте результат каждого шага.
- Поощряйте самостоятельность — пусть дети пробуют менять числа и тексты.
- Используйте визуальные подсказки и эмодзи для наглядности.
- Всегда проверяйте вместе, объясняйте ошибки — ошибки — это нормально!

Тема 9-10: Защита проекта, рефлексия, повторение (для детей с ООП)

Цель обучения: записывать линейные алгоритмы на языке Python (пайтон)

Теория:

- Принципы презентации результата:
 - просто и понятно рассказать о своей программе,
 - показать, как она работает.
- Как рассказать о своей программе:
 - что спрашивает, что считает, что выводит.

Практика:

- Защита проекта в удобной форме: устно, письменно, демонстрация кода.
- Обсуждение трудностей и успехов.
- Выдача индивидуальных рекомендаций.

Цель урока:

- Показать, что умеешь делать программу.
- Обсудить, что получилось, а что можно улучшить

Задания для учеников:

- Расскажи или покажи, как работает твоя программа.
- Нарисуй схему или расскажи словами, что делает программа по шагам.
- Ответь на вопросы:
 - «Что было интересно?»
 - «Что получилось?»
 - «Что хочешь улучшить?»

Таблица критериев оценки (10 баллов)

Критерий	Баллы	Комментарий
Планирование проекта (цель, ввод, логика)	2	Программа имеет чёткий план и логику
Использование переменных	2	Переменные названы понятно и используются верно
Ввод данных (input())	1	Данные вводятся корректно
Преобразование типов (int(), float())	1	Правильно используется преобразование
Арифметические операции	2	Верно выполнены вычисления
Вывод результата (print())	1	Результат показан понятно
Комментарии в коде	1	Код содержит понятные комментарии
Итого	10	

Общие методические рекомендации для педагога (дети с ООП)

- Поддерживайте дружелюбную и спокойную атмосферу, обеспечивайте эмоциональную безопасность.
- Используйте адаптированные среды программирования (например, Thonny с крупным шрифтом и подсветкой кода).
- Обеспечивайте гибкий темп обучения: давайте больше времени, избегайте спешки.
- Учитывайте индивидуальные особенности и рекомендации ПМПК.
- Завершайте каждый урок рефлексией: смайлики, шкала «понравилось — не понял», обсуждение успехов.

Карточки для урока «Мини-проект: написание простой программы»

Карточка 1: Переменные и ввод данных

Переменная — это "коробочка" для хранения данных.

```
python
```

```
имя = input("Как тебя зовут? ")  
возраст = int(input("Сколько тебе лет? "))
```

- `input()` — спрашивает у пользователя текст.
- `int()` — превращает текст с цифрами в число.

Карточка 2: Вывод данных на экран

Вывод — команда, которая показывает результат пользователю.

```
python
```

```
print("Привет,", имя + "!")  
print("Тебе уже", возраст, "лет.")
```

Карточка 3: Арифметические операции

В Python можно складывать, вычитать, умножать и делить числа:

```
python
```

```
сумма = 5 + 3  
разность = 10 - 2  
произведение = 4 * 7  
частное = 20 / 4
```

Карточка 4: Планирование программы

1. **Цель:** Что программа должна делать?
2. **Ввод:** Какие данные спросить у пользователя?
3. **Логика:** Что считать или делать с этими данными?
4. **Вывод:** Что показать в конце?

Карточка 5: Пример программы «Приветствие»

python

```
имя = input("Как тебя зовут? ")
возраст = int(input("Сколько тебе лет? "))
print("Привет,", имя + "!")
print("Тебе уже", возраст, "лет.")
```

Карточка 6: Пример программы «Магазин»

python

```
цена = float(input("Введите цену товара: "))
количество = int(input("Введите количество товара: "))
итого = цена * количество
print("Общая стоимость:", итого)
```

Карточка 7: Советы по защите проекта

- Расскажи, что делает твоя программа.
- Покажи, как она работает.
- Объясни по шагам логику программы.
- Поделись, что было интересно и что получилось.