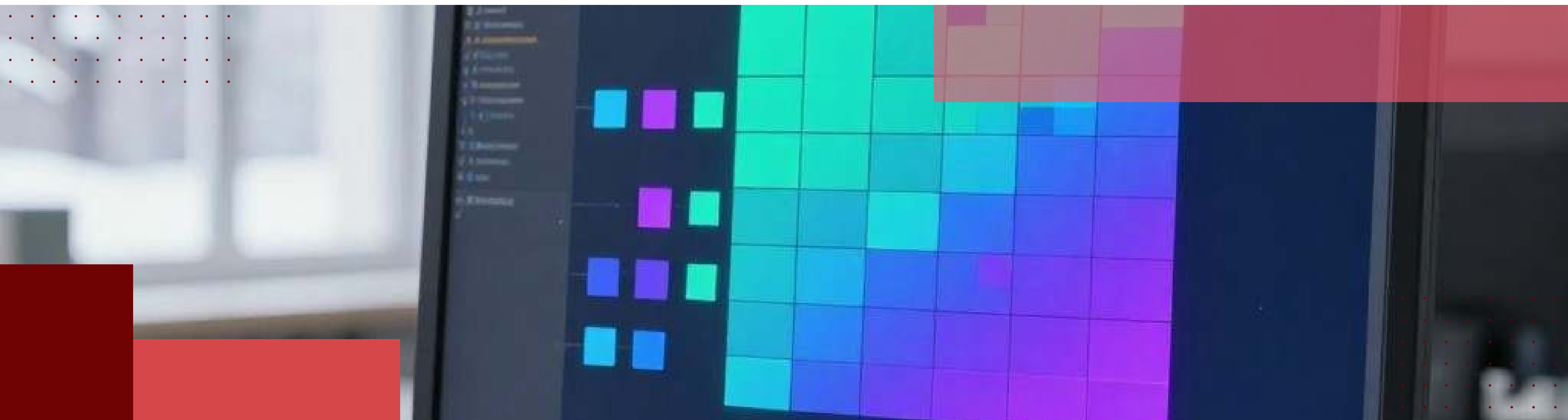


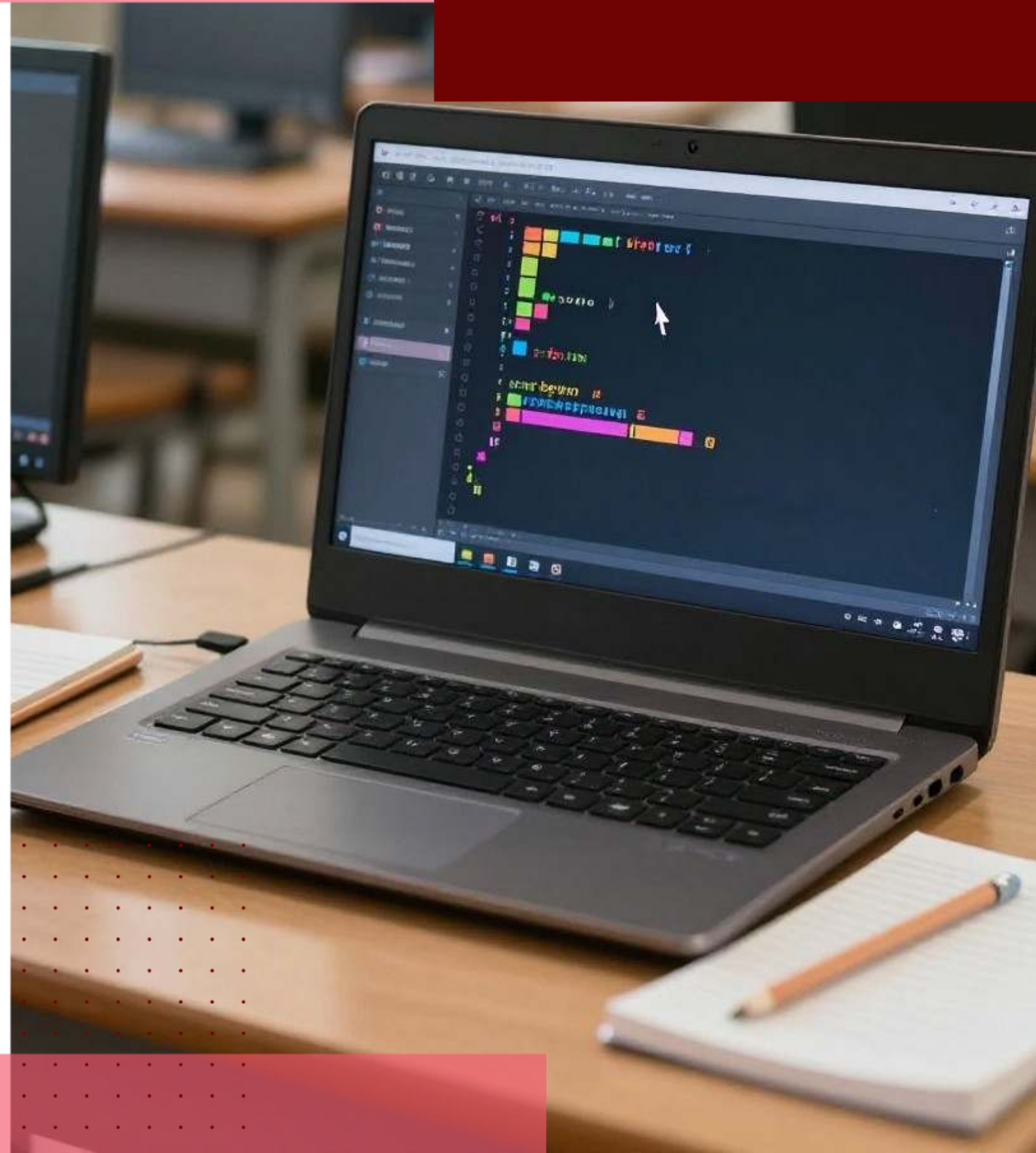
# Первая программа на языке Паскаль

Паскаль помогает освоить команды, переменные, ввод, вывод и простые вычисления. Программа оживает на экране.



# С чего начинается программирование в 8 классе

Первая программа знакомит с алгоритмом и правилами записи команд. На Паскале особенно хорошо видно структуру, служебные слова и порядок действий. Школьники осваивают программу, команду, переменную, тип данных, компиляцию и выполнение.



# Краткая история языка Паскаль

## 1970 год — создание языка

Никлаус Вирт разработал Паскаль как учебный язык для обучения структурному программированию и аккуратной записи программ.

## 1980–1990-е — популярность в средах разработки

Turbo Pascal и Delphi получили широкую известность, и Паскаль часто использовали в школах и на первых курсах программирования.

## Сильная сторона — строгая структура

Язык ценят за понятные правила оформления и ясную структуру, благодаря которой начинающим проще не запутаться в коде.

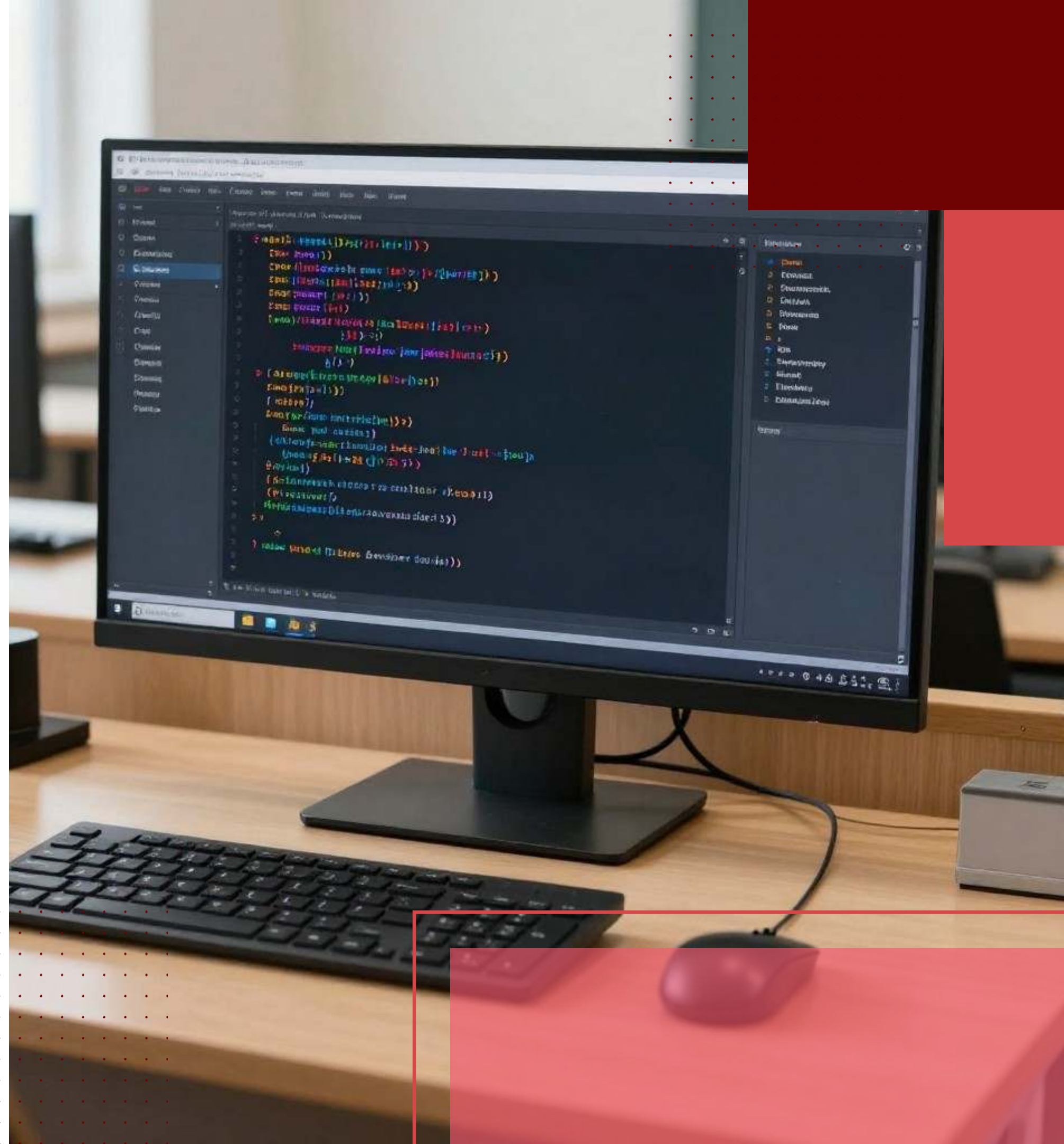
# Среда, где пишут и запускают программу

## Где создают код

Для работы с Паскалем используют среды программирования Free Pascal, Lazarus или Turbo Pascal. Исходный код обычно сохраняют в файле с расширением .pas. Так текст программы становится «рабочим материалом».

## Что происходит при запуске

Во время запуска видны редактор с кодом, сообщения компилятора и кнопка старта. Компилятор проверяет синтаксис, а среда показывает путь от написанного текста к результату на экране.



# Структура самой простой программы

Минимальная программа начинается с заголовка `program`, затем идёт блок описаний, а после него — раздел `begin...end`. Такая схема помогает сразу понять, где стартует программа и где заканчиваются команды.

Внутри `begin...end` размещаются операторы, которые выполняются по порядку сверху вниз. Именно последовательность команд формирует результат, даже если в программе всего одна операция.

После слова `end` обязательно ставится точка. Это важное правило: оно показывает завершение программы и помогает избежать ошибок компиляции из-за мелких знаков.

# Основные элементы первой программы

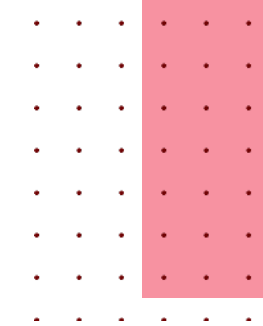
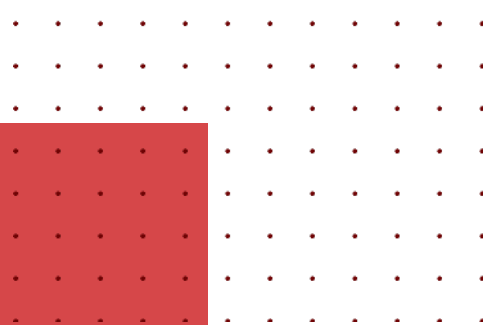


Таблица кратко показывает назначение ключевых слов и правила записи знаков в учебном Паскале.

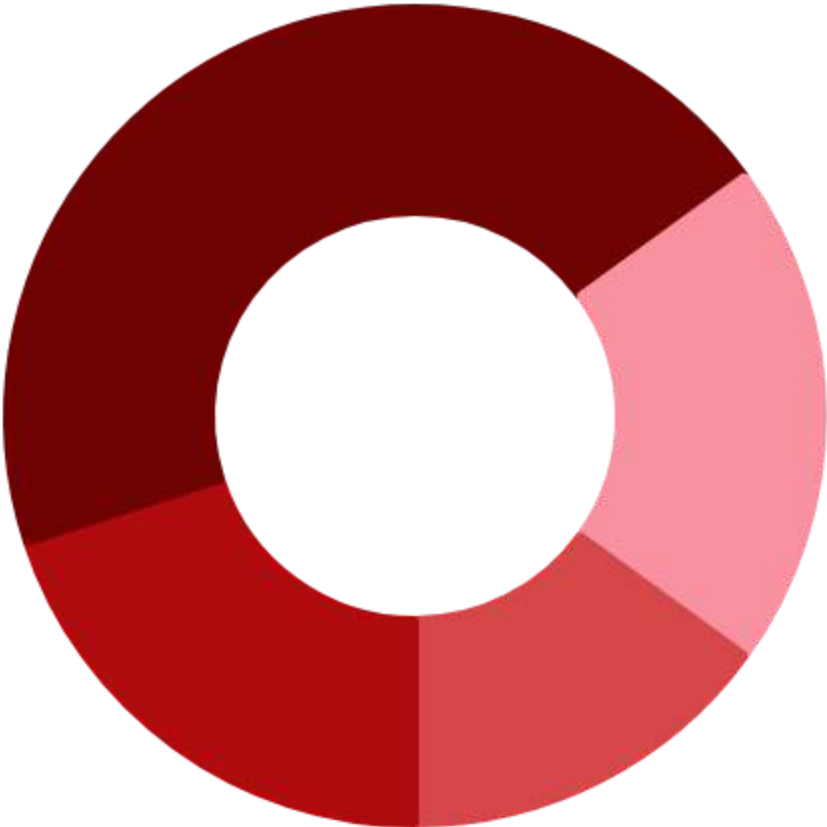
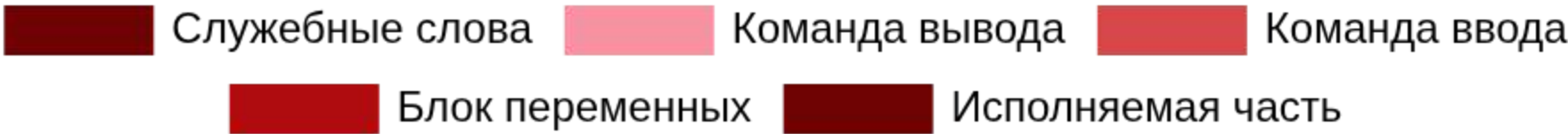
Правильная запись ключевых слов и знаков (в том числе точка после end.) напрямую влияет на то, компилятор успешно выполнит программу.

| Элемент языка | Пример записи     | Назначение               |
|---------------|-------------------|--------------------------|
| program       | program Name;     | Имя программы            |
| var           | var a: integer;   | Объявление переменных    |
| begin         | begin             | Начало исполняемой части |
| writeln       | writeln('текст'); | Вывод строки на экран    |
| readln        | readln(x);        | Ввод значения            |
| end.          | end.              | Завершение программы     |

Школьный курс информатики: базовые конструкции языка Паскаль



# Из чего состоит первая программа



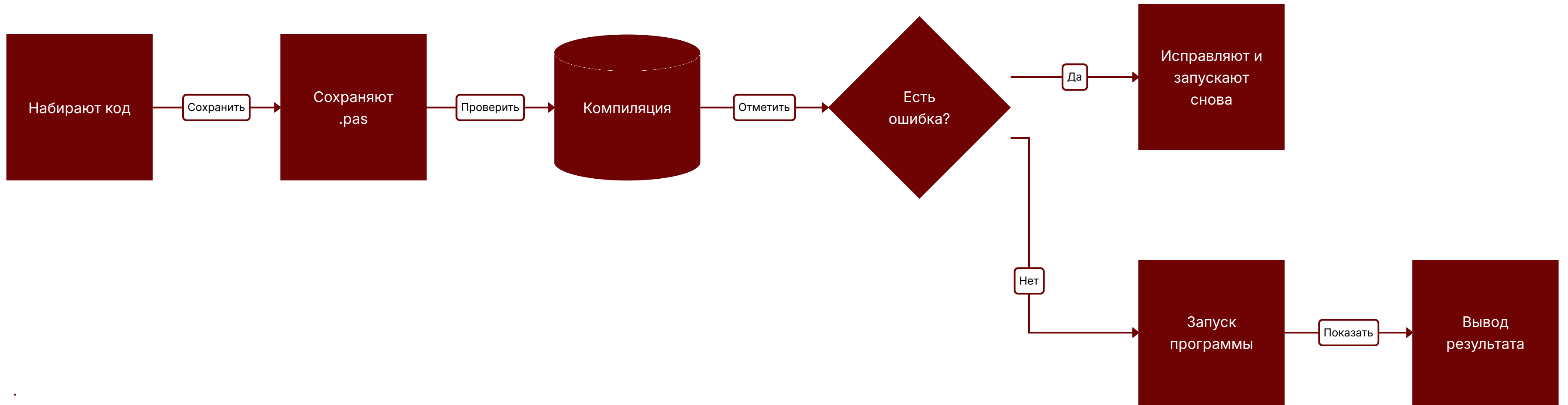
Обобщённая модель учебной программы на Паскале (без привязки к конкретной среде)

Даже в коротком примере выполняемая часть играет главную роль: именно она управляет тем, что увидит ученик в результате.

Визуальная структура показывает: программа может быть небольшой, но её «скелет» всё равно задают ключевые части — от словаря языка до блока команд.

# Путь от набора текста до результата

Последовательность действий при компиляции и запуске программы на Паскале



# 1

**Именно это простое действие делает результат видимым без вычислений и ввода.**

Учебный пример первой программы на Паскале (конструкция `writeln`)

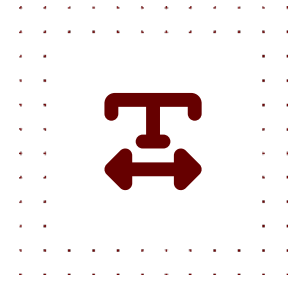
Команда `writeln` выводит строку на экран: текст берётся в одинарные кавычки. Самое главное — ученик сразу видит сообщение после запуска.

# Переменные и их назначение



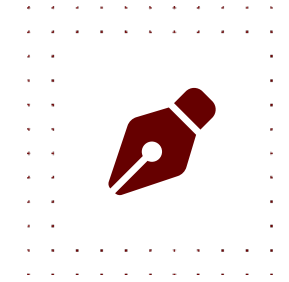
## Память для значений

Переменная хранит значение в памяти компьютера и может использоваться в программе несколько раз. Это помогает связывать ввод пользователя и дальнейшие действия с данными.



## Типы данных для школьных задач

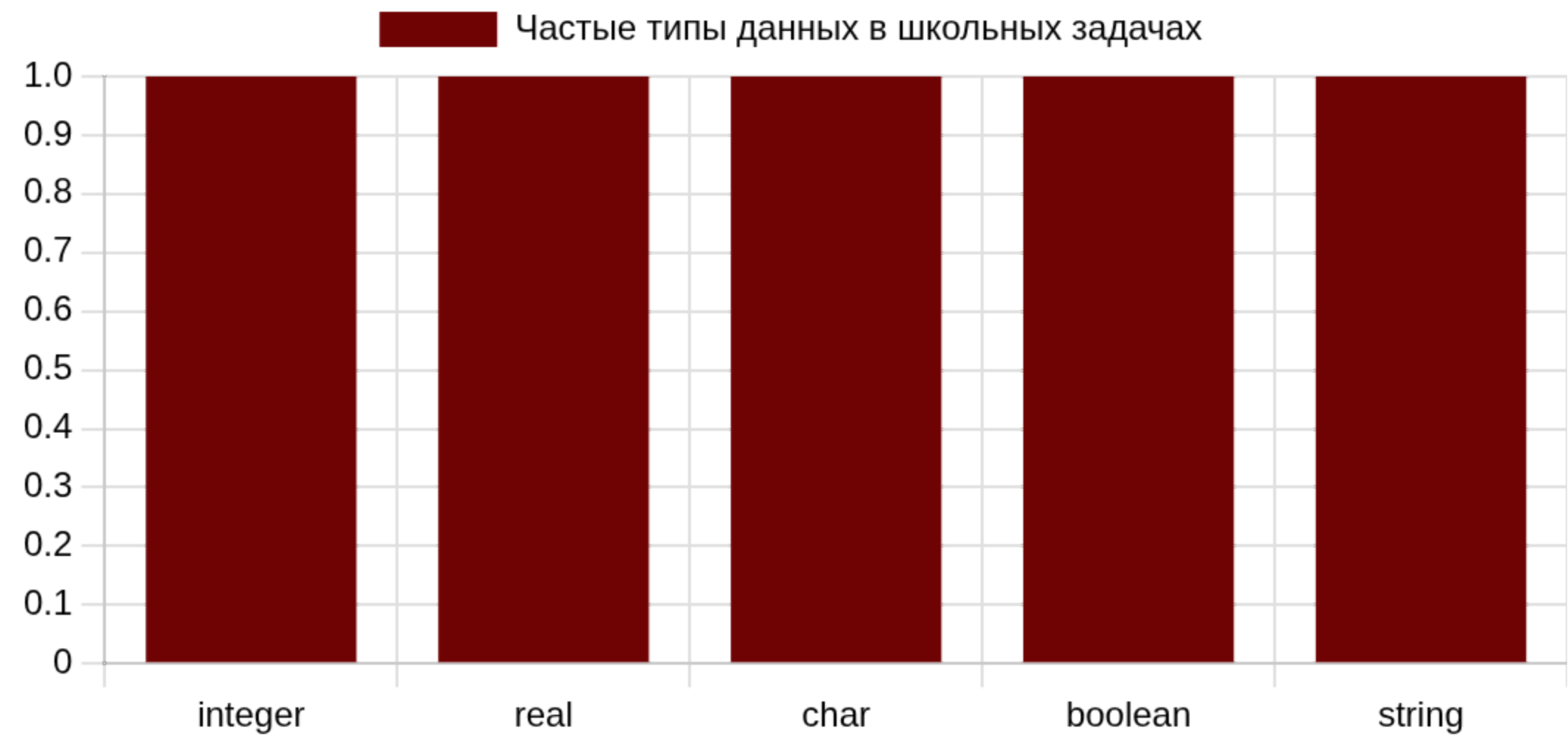
Для первых заданий чаще всего подходят целые числа `integer` и текст `string`. Тип данных задаёт, какие значения разрешены, и помогает компилятору проверять правильность операций.



## Объявление в разделе `var`

Переменные объявляют после слова `var`. Например: `var a: integer; s: string;`. Так программа «знает», какие имена и форматы данных будут встречаться в командах.

# Частые типы данных в школьных примерах



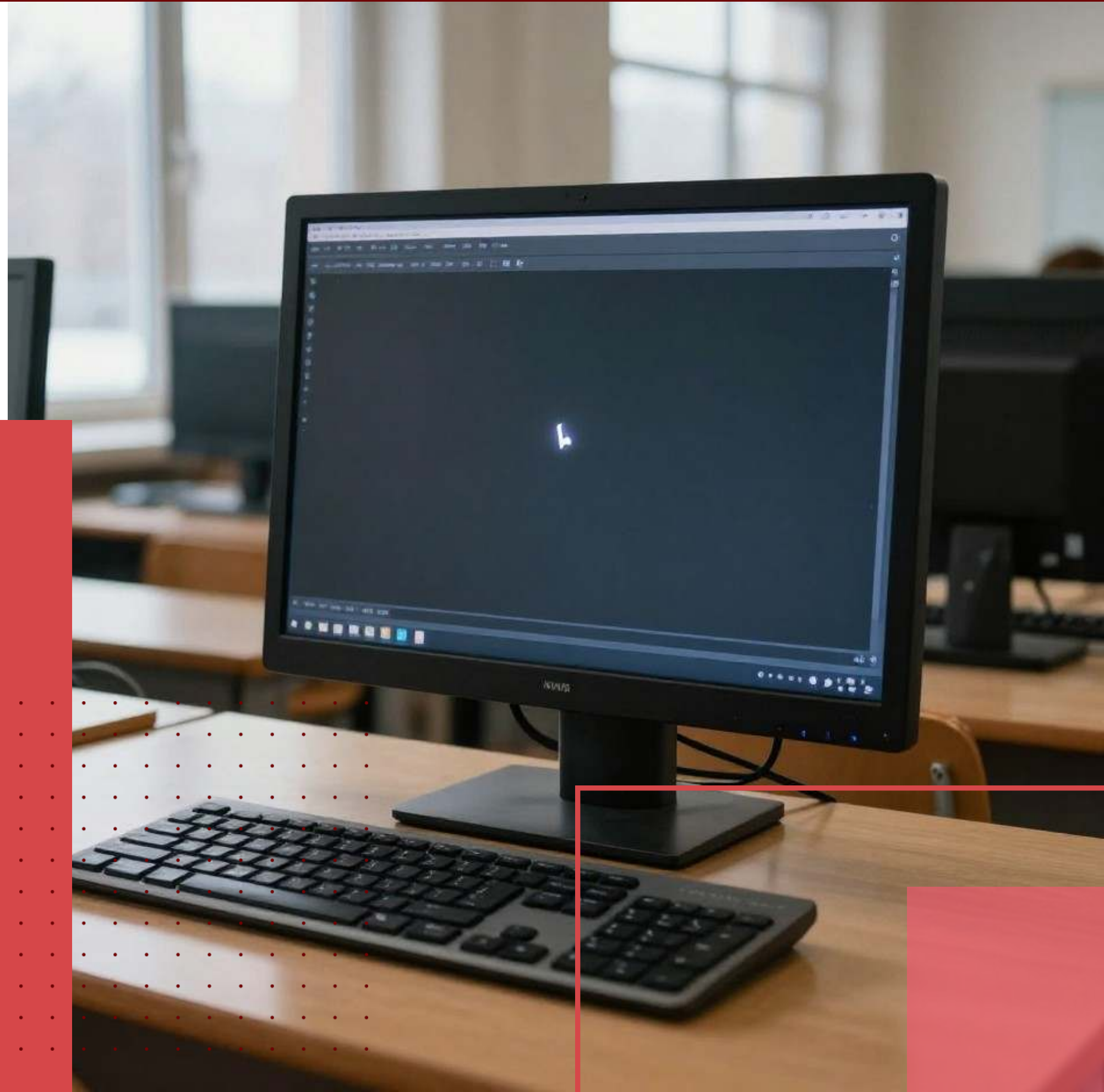
На практике тип данных выбирают под задачу: для целых чисел удобен `integer`, для текста — `string`. Другие типы помогают уточнить, какие значения разрешены.

Диаграмма показывает, что стартовый набор типов невелик. Это упрощает первые программы: ученику не нужно угадывать формат данных.

Учебные сведения по типам данных Паскаля (методические материалы курса информатики, без количественных измерений).

# Ввод данных с клавиатуры

1. Команды `read` и `readln` считывают значения, которые вводит пользователь с клавиатуры. Обычно сначала в переменную попадает то, что пользователь набрал в ответ на запрос программы. Затем можно сразу использовать это значение в вычислениях.
2. В учебных заданиях чаще вводят одно число, имя или несколько значений подряд. Удобно связать ввод с кодом: курсор мигает в поле ввода, а в окне компилятора видно, что программа получила введённые данные.



# Простейшие вычисления в Паскале

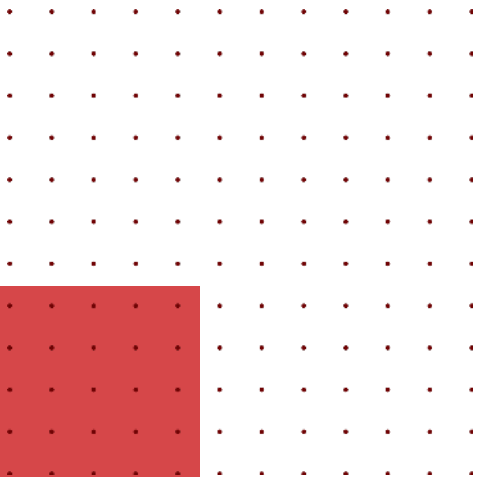


Паскаль умеет выполнять базовые арифметические операции: сложение, вычитание, умножение и деление чисел. Для первых задач обычно берут две величины и получают новое число.

Частые примеры — сумма двух чисел, разность и произведение. В таких заданиях важно, чтобы переменные были правильно объявлены и соответствовали типу исходных данных.

Иногда рассматривают формулу площади прямоугольника:  $a*b$ . Здесь выражение записывают в одну строку, а результат сразу выводят на экран.

Чтобы результат был верным, следите за порядком действий. Если нужно изменить стандартный порядок, применяйте скобки, например,  $(a+b)/2$ , а не  $a+b/2$ .



# Типичные ошибки начинающих программистов

| Ошибка в коде              | Как выглядит                      | Частая причина           |
|----------------------------|-----------------------------------|--------------------------|
| Пропущена точка после end. | end; вместо end.                  | Синтаксис                |
| Забыта точка с запятой.    | Нет ; после оператора.            | Символы и правила записи |
| Неверно имя переменной     | var a: integer; а в коде b.       | Внимательность           |
| Неправильные кавычки       | writeln(Привет); вместо 'Привет'. | Синтаксис                |
| Перепутан тип при вводе    | в integer вводят текст.           | Несоответствие типов     |

Типовые трудности учащихся при изучении Паскаля (обобщение практики выполнения стартовых работ).



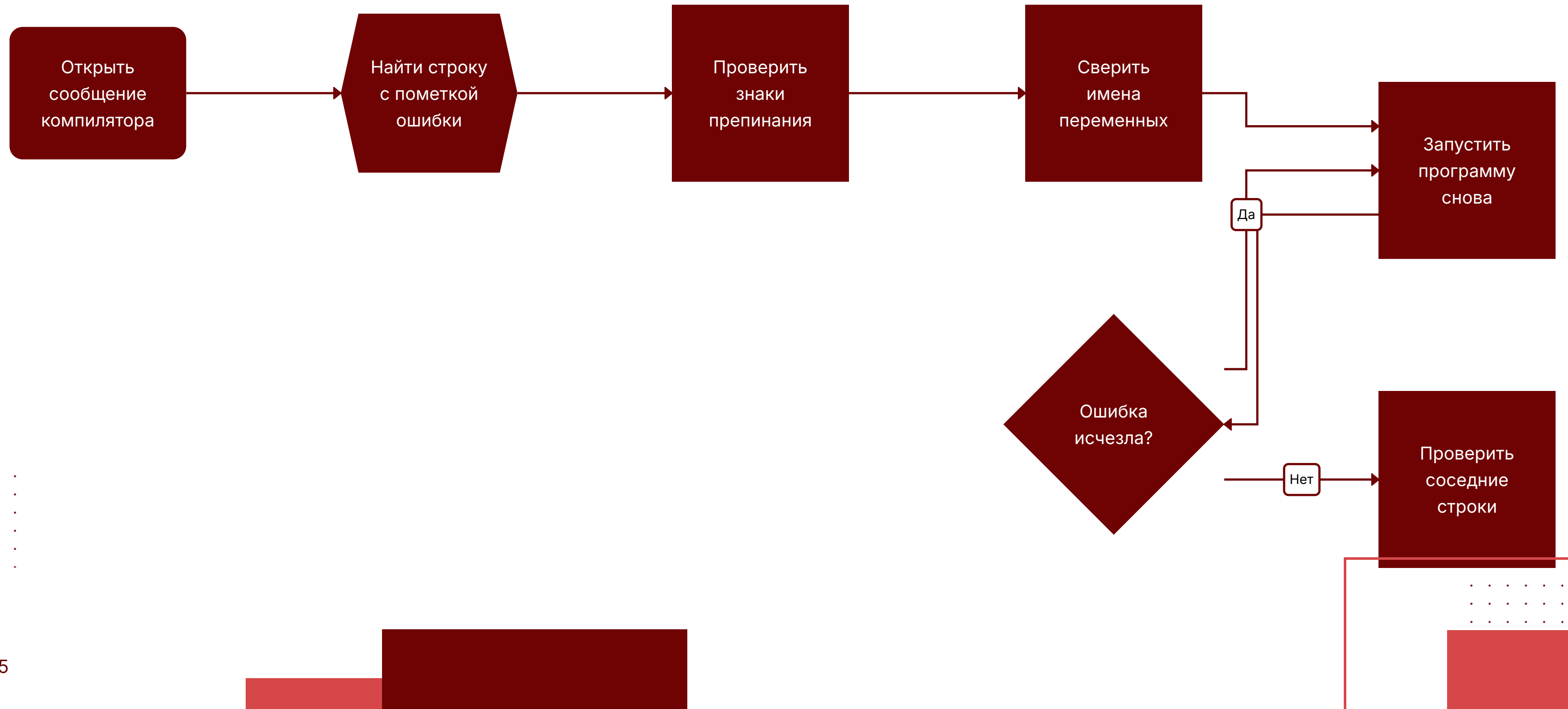
В таблице собраны частые промахи при написании первых программ и кратко указаны причины: синтаксис, внимательность к символам или несоответствие типов.

Большинство проблем связано не с логикой, а с точностью записи. Быстрая проверка знаков и имён обычно помогает исправить ошибку за минуту.



# Как найти и исправить ошибку в коде

От сообщения компилятора до повторного запуска программы



# Пример учебной программы с именем и возрастом

## Шаг 1 — запрос данных

Программа выводит на экран вопросы и ожидает ввод. Пользователь вводит имя и затем возраст с клавиатуры, чтобы дальше обработать эти значения.

## Шаг 3 — формирование сообщения

Программа собирает фразу с именем и возрастом, используя переменные. Текст сообщения подбирается так, чтобы ученик сразу понимал, что именно вывел компьютер.

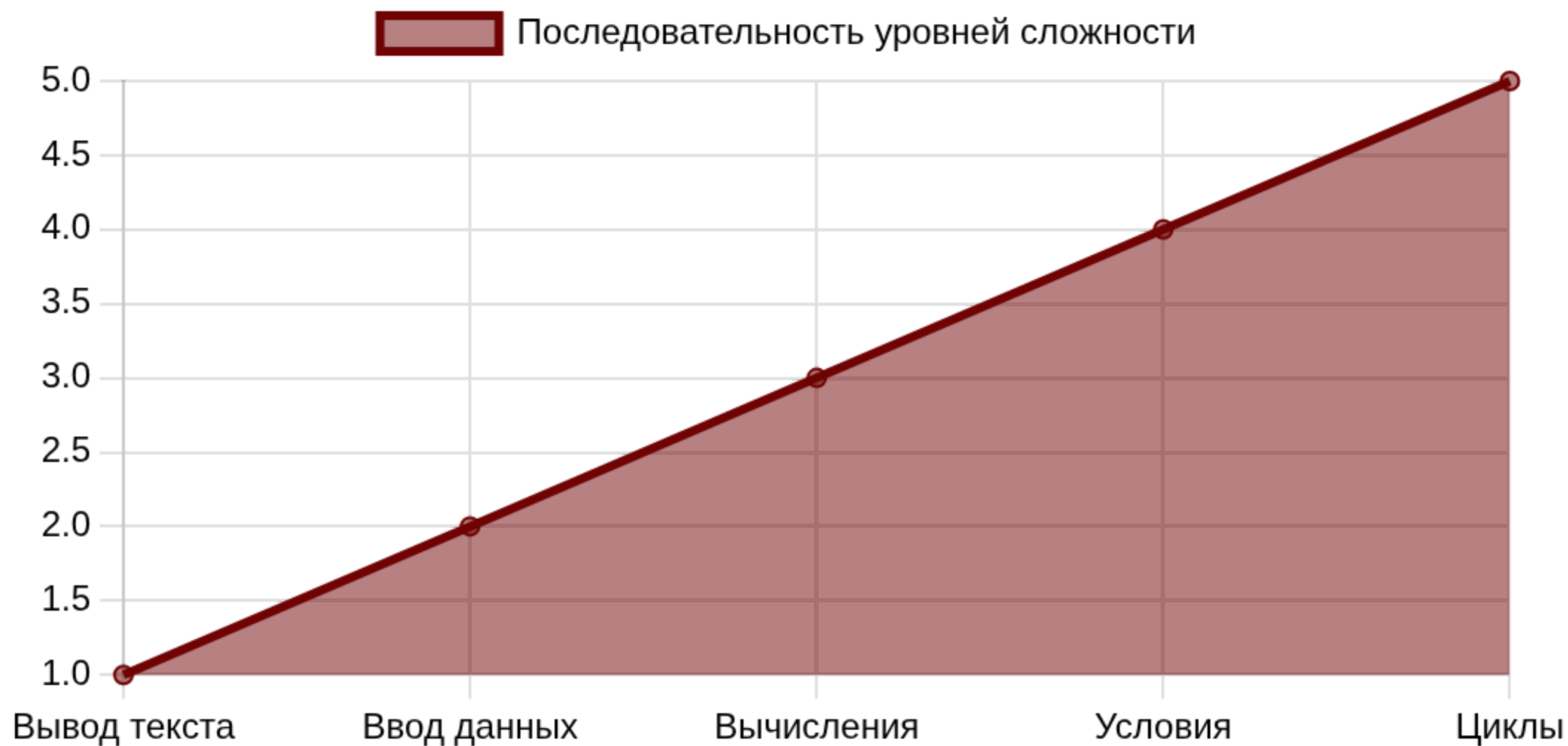
## Шаг 2 — сохранение в переменных

Ввод сохраняется в переменных: строка для имени и число для возраста. Так данные становятся доступными для последующих команд, не нужно повторно вводить их.

## Шаг 4 — вывод результата

Наконец программа печатает готовое приветствие или короткое описание. Результат появляется в окне консоли, и ученик видит связь между вводом и выводом.

# Уровни сложности первых задач



Линейная шкала помогает увидеть «лестницу» заданий: от вывода сообщений к обработке данных, затем к ветвлениям и повторениям.

Первые программы становятся опорой. Освоив структуру и ввод-вывод, проще понять, как условия и циклы управляют ходом работы.

Учебная последовательность тем в курсе информатики (типовая структура практических работ).

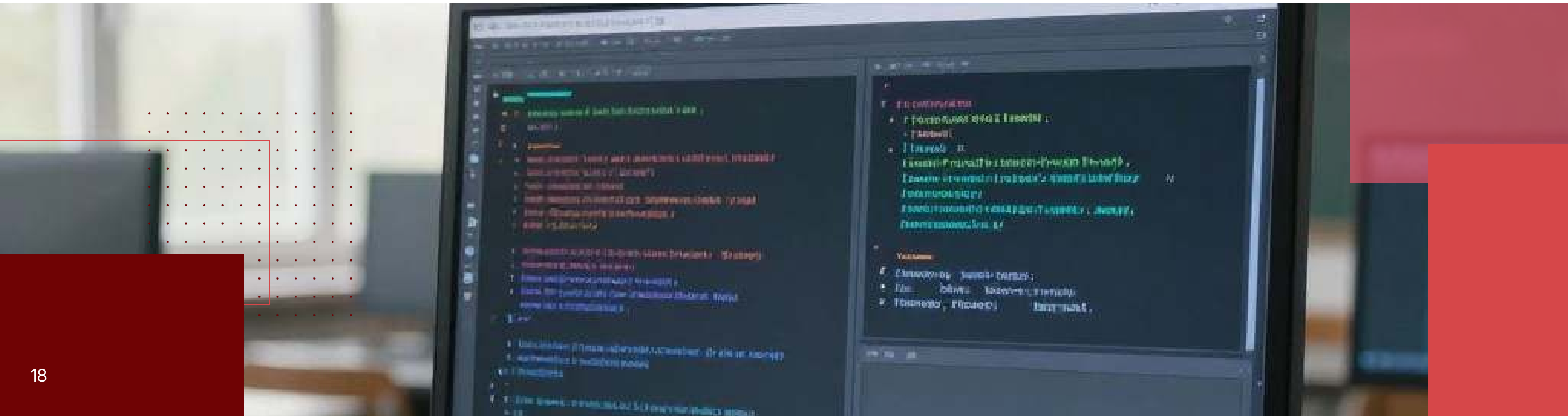
# Как выглядит результат выполнения программы

## Где появляется вывод

После запуска программы результат выводится в окне консоли или в специальном окне среды программирования. Там отображается всё, что программа успела выполнить до завершения.

## Что именно показывается

Вывод может быть приветствием, числом после вычислений, обработанным значением из ввода или фразой с именем пользователя. По нему ученик понимает, какие строки кода сработали.



# Чему учит первая программа на Паскале

Первая программа формирует аккуратность: нужно писать код точно, по правилам и без лишних символов. Даже небольшая ошибка в знаках мешает выполнению, поэтому дисциплина закрепляется быстро.

Школьник осваивает структуру программы, понимает назначение её частей и учится читать код сверху вниз. Он получает представление о роли служебных слов, блоков и завершающих признаков.

Также отрабатываются ввод и вывод данных: переменные помогают хранить значения, а команды позволяют показать результат. Эти навыки пригодятся при изучении Python и C++ — меняется синтаксис, логика остаётся.

# Главный вывод по теме первой программы

Первая программа на Паскале наглядно показывает, как текстовый алгоритм превращается в работающий результат. Для 8 класса это понятный старт благодаря простой структуре, строгим правилам и быстрому ответу после запуска.