

Языки программирования

Основные структурные элементы • План учителя • 10 класс РБ • 45 минут

START: подготовка и границы

Второй урок, базовый уровень. Основа: Котов В. М. и соавторы, «Информатика. 10 класс», 2020, § 2, с. 11–21. Веб-урок объясняет тему ученику; этот документ содержит сценарий, условия, ключи и помощь учителю. Презентация — опора для совместного разбора, а не текст для сплошного переписывания.

Класс: 26 человек. Нужны доска, тетради, ручки и часы. Проектор желателен. Печать и компьютеры не обязательны. Первое знакомство с комплектом: 10–15 минут; повторная подготовка: около 5 минут. Откройте презентацию. Без проектора заранее выпишите определение, программу со страницы 3 и условия А–В; подробности на странице 6. Это ещё 7–10 минут подготовки доски, итого 17–25 минут при первом знакомстве. При готовом PascalABC.NET проверьте запуск заранее: установка и настройка в 45 минут не входят.

Чему учим и как увидим результат

Ученик объяснит отличие алгоритма от программы, узнает имя и тип переменной, выражение и оператор, проследит выполнение и изменит готовую программу.

Свидетельства: личные ответы А, объяснение Б3, полная программа площади В с двумя тестами и выходной билет. Ключи и критерии на страницах 4–5. Обзор уровней языков, трансляции и парадигм помогает читать § 2; создавать классы и собственные функции сегодня не требуется.

Маршрут на 45 минут

0–3 связь с алгоритмом; 3–6 определение и уровни; 6–8 выполнение; 8–10 синтаксис/семантика; 10–12 имена; 12–15 переменные и присваивание; 15–23 программа, трассировка, ошибки; 23–25 подпрограммы; 25–28 обзор парадигм; 28–31 А; 31–34 Б; 34–40 В; 40–43 билет; 43–45 домашнее. Время попытки, сверки и переходов включено.

Предпосылки

Алгоритм и массовость из урока 1; сложение и умножение целых чисел. Повторите: периметр прямоугольника — сумма четырёх сторон, площадь — произведение двух соседних сторон. Во всех основных программах стороны целые от 1 до 1000 см. Это границы примеров, а не универсальное ограничение языка.

Объяснение до самостоятельной работы

0–6 минут. Слайды 1–3

Спросите: «Как найти периметр при сторонах 5 и 3 см?» Сумма 8, удвоение 16. Скажите: «Способ решения нам известен; теперь запишем его для компьютера». Дайте определение: язык программирования — формальный язык для записи программ, задающий правила записи команд и данных и их смысл. Программа — запись действий и данных средствами языка, предназначенная для выполнения компьютером. На определение и запись выделите 1,5 минуты, на объяснение и уровни ещё 1,5.

Высокий уровень позволяет не расписывать каждую машинную команду: Pascal, Python, C++, Java. Низкий уровень тесно связан с процессором: ассемблер. «Высокий» не означает «лучший». Один алгоритм может иметь записи на разных языках.

6–10 минут. Слайды 4–5

Транслятор преобразует программу из одного языка или представления в другое. Компилятор переводит в иную форму, например машинный или промежуточный код, который затем выполняют. Интерпретатор обрабатывает команды или промежуточное представление во время работы. Способы могут сочетаться. PascalABC.NET — среда с редактором и компилятором, Pascal — язык. Синтаксис означает правила записи; семантика — смысл. Прочитайте $p := 2 * (a + b);$: сложить a и b , удвоить, записать в p .

10–15 минут. Слайды 6–8

Алфавит: буквы, цифры, знаки, разделители. Служебные слова `var`, `begin`, `end` имеют заданную роль. Идентификатор — имя объекта. В наших примерах первая — латинская буква, затем буквы, цифры или подчёркивание. `side2` и `total_sum` подходят; `2side` и `begin` — нет. Переменная — именованная величина с изменяемым значением. Тип определяет значения и операции. Константа обозначает постоянное значение.

Для программы нужен `integer` — целые числа. Обзор других типов без диктовки: `real` — вещественные приближённые значения; `boolean` — `True/False`; `char` — символ; `string` — строка. Покажите число 10 и текст '10'. Выражение $2 * (a + b)$ вычисляет значение. Оператор $p := 2 * (a + b);$ сохраняет его. $:=$ — присваивание. При старом $x = 4$ команда $x := x + 1;$ записывает 5. $+$ — операция, весь оператор присваивания — законченное действие.

Готовая программа и её выполнение

15–23 минуты. Слайды 9–11

Данные — целые a и b от 1 до 1000 см, результат — периметр в сантиметрах. Четыре минуты на чтение и запись шести строк:

```
var a, b, p: integer;
begin
  readln(a, b);
  p := 2 * (a + b);
  writeln(p);
end.
```

`var` описывает переменные и их тип; `begin` начинает блок; `readln` получает два числа; `p := ...` вычисляет и сохраняет ответ; `writeln` выводит значение; `end.` заканчивает программу. Точка с запятой разделяет операторы, точка завершает программу. Умножение пишется явно: $2 * (...)$.

Две минуты на трассировку: ввод 5 3 $\rightarrow a = 5, b = 3 \rightarrow$ сумма 8 $\rightarrow p = 16 \rightarrow$ вывод 16. Затем 4 2 $\rightarrow 12$. Код тот же, данные и ответ иные. Если среда заранее готова, второй прогон можно заменить запуском: в окне ввода 5 3 и Enter. Без компьютера значения записываются в тетради.

Две минуты на ошибки: `p := 2 * ;` — недостаёт выражения; `p := 'hello';` — строка не подходит `integer`; `p := a + b;` — допустимая запись, но неверная формула периметра. Компилятор обнаружит первые две ошибки; последняя требует сравнения с ожидаемым ответом 16, а не 8.

23–28 минут. Слайды 12–13

Подпрограмма выполняет отдельную задачу и вызывается по имени. Функция `Abs(-3)` возвращает 3, это значение можно использовать в выражении. Процедура `writeln(p)` выводит число, не возвращая значение как результат вызова. Процедуры могут изменять данные; не говорите, что у них вообще нет результата.

За три минуты дайте обзор § 2.2: структурный подход — следование, ветвления, циклы, подпрограммы; процедурный — разбиение действий на процедуры и функции; функциональный — вычисление и сочетание результатов функций; объектный — объекты с состоянием и поведением. Пример объекта: кнопка и реакция на нажатие. Это парадигмы организации программы, которые могут сочетаться. Практическое освоение всех четырёх сегодня не проверяем.

Самостоятельные задания и ключи

28–31 минута. А, слайды 14–15

Оставьте на экране условие А, ученики открывают записанный образец в тетради. Без проектора код и условия на доске. Каждый записывает: 1) язык, переменные, тип, строки ввода и вывода; 2) сумму и вывод для 6 3; 3) выражение и оператор присваивания. 20 секунд инструкция, 2 минуты попытка, 40 секунд сверка. Ключ: Pascal; a, b, p: integer; readln(a,b); и writeln(p); сумма 9, вывод 18; выражение $2 * (a + b)$, оператор $p := 2 * (a + b);$. При ответе 9 подчеркните удвоение, новый вход $2\ 3 \rightarrow 10$.

31–34 минута. Б, слайды 16–17

1) Какие имена подходят: side2, 2side, begin, total_sum? 2) Почему при p: integer не подходит $p := 'hello';$? 3) Для периметра записали $p := a + b;$. Ввод 6 2: фактический и ожидаемый ответы, примет ли компилятор? Время: 20 секунд инструкция, 2 минуты попытка, 40 секунд сверка. При задержке А вопросы Б1–2 обсудите устно, Б3 оставьте каждому письменно.

Ключ: side2 и total_sum подходят; 2side начинается с цифры, begin занято. Строка не соответствует integer. Допустимая строка даст 8, а нужно 16. Исправить на $p := 2 * (a + b);$, проверить $3\ 3 \rightarrow 12$. При путанице имён предложите side_length; при путанице выражения и оператора предложите $p := p + 2;$ при старом $p = 10 \rightarrow 12$.

34–40 минут. В, слайды 18–19

Найти площадь для целых сторон от 1 до 1000 см: сохранить ввод, результат назвать area, изменить описание, вычисление и вывод. Написать полную программу; проверить 5 3 и 4 2; объяснить согласование имени. 30 секунд инструкция, 3,5 минуты запись, 1,5 минуты сверка, 30 секунд исправление. С компьютером меняют готовый код и запускают; без него пишут в тетради.

```
var a, b, area: integer;
begin
  readln(a, b);
  area := a * b;
  writeln(area);
end.
```

Ключ: вывод 15 и 8, единицы по условию см². area описана, получает вычисленное значение и выводится. Быстрым: $3\ 3 \rightarrow 9$. Затрудняющимся дайте первую строку, вычисление ученик дописывает сам. Имя p во всех трёх местах допустимо при верной формуле, но area яснее передаёт смысл.

Оценка понимания и завершение

40–43 минуты. Выходной билет, слайд 20

1) Чем программа отличается от алгоритма? 2) Было $x = 4$. Что сделает $x := x + 1$? 3) Доказывает ли успешная компиляция верный ответ? Объясни на примере периметра. Время: 20 секунд инструкция, 1 минута 40 секунд запись, 40 секунд сверка одного примера, 20 секунд подготовить тетради к сдаче.

Ключ: алгоритм задаёт способ решения, программа записывает действия и данные на языке программирования; x станет 5; нет, строка $a + b$ может компилироваться, но не вычисляет периметр. Пересказ своими словами допустим. Соберите билет вместе с личной программой площади.

Критерии и дальнейшие действия

Программа В: описаны используемые переменные; сохранён ввод; формула $a * b$; выводится вычисленная площадь; оба прогноза верны. Билет: различает способ и его запись; понимает замену значения; отличает компиляцию от проверки результата. Отмечайте компоненты отдельно: освоено / нужна помощь, без автоматического перевода в отметку.

После урока просмотрите записи (ориентир 8–12 минут на класс). Если неверна формула, верните к смыслу площади и новому входу 3 3. Если осталась старая переменная в выводе, проследите, где хранится вычисленное значение. Если $:=$ трактуется как равенство, покажите последовательность $x = 4 \rightarrow$ вычислили 5 $\rightarrow$ заменили x . Если смешаны язык и среда, сопоставьте Pascal и PascalABC.NET. На следующем занятии проверьте коротким новым примером именно найденное затруднение.

43–45 минут. Домашнее задание, слайд 21

Прочитать § 2, особенно 2.3. Написать программу периметра квадрата: целая сторона от 1 до 1000 см, имена a и $perimeter$, тип `integer`. Подписать ввод, выражение, вывод. Проверить 3 см и 5 см. Компьютер необязателен. На запись условия — минута, повтор условия — 30 секунд, сбор тетрадей — 30 секунд.

```
var a, perimeter: integer;
begin
  readln(a);
  perimeter := 4 * a;
  writeln(perimeter);
end.
```

Ключ: 3 $\rightarrow$ 12, 5 $\rightarrow$ 20, периметр в сантиметрах. Если ученик складывает четыре стороны вместо умножения, такой вариант вычисления тоже верен.

Варианты проведения и источники

Без техники и при сокращении времени

Без проектора заранее выпишите на доске определение, программу периметра и условия А–В; ориентир подготовки 7–10 минут. Остальные объяснения полностью приведены на страницах 2–3. Без принтера и телефонов ученики работают в тетрадях. Чтение и изменение программы не зависят от наличия компьютера.

Если класс медленно переписывает, не диктуйте все типы и парадигмы: они есть в веб-уроке для повторения. Сохраняйте явное определение, полный пример, индивидуальную программу площади и билет. При готовой компьютерной среде запуск заменяет часть устной трассировки, а не добавляется к маршруту.

40 минут: связь и определение 5; выполнение и синтаксис 3; имена, типы и присваивание 5; полный пример 7; подпрограммы и обзор парадигм 4; А 3; Б 3; В 5; билет 3; домашнее 2. Всего 40. Сократите устные примеры и число ответов вслух; Б1–2 можно устно. Не сокращайте личную попытку до показа ключа. Для В за 5 минут: 30 секунд инструкция, 3,5 минуты личная попытка, 30 секунд сверка двух ответов и трёх изменённых строк, 30 секунд исправление. Развёрнутое объяснение проверяется после урока по записи ученика.

Источники и авторские решения

Котов В. М., Лапо А. И., Быкадоров Ю. А., Войтехович Е. Н. «Информатика. 10 класс». Народная асвета, 2020, § 2, с. 11–21. Параграф охватывает уровни языков, средства выполнения, парадигмы и структурные элементы. Определения пересказаны своими словами. Код, задачи и ключи созданы для этого урока; сканы учебника не распространяются.

Программа X–XI классов, постановление от 29.07.2025 № 132: <https://adu.by/images/2025/08/12/Inform-10-11.pdf>. При аудите курса подтверждено повторение языка программирования в начале X класса. Тема урока следует принятому КТП, § 2.

Официальная справка PascalABC.NET:

<https://pascalabc.net/downloads/pabcnethelp/index.htm>. Для пояснения методов выполнения учтено, что компиляция и интерпретация могут сочетаться. Переход «прочитать готовый код → проследить → изменить» использует принцип моделирования решения: <https://teachcomputing.org/pedagogy>. Обзор парадигм ограничен знакомством, основная практика — данные и операторы.

Четыре оригинальные иллюстрации image_gen, 15.09.2026: язык, трансляция, переменные, подпрограммы. Это образы понятий; точные команды и значения даны редактируемым текстом. Материал требует обычной учительской адаптации к классу; реальная классная апробация не проводилась.