

**Беспилотники и
нейросети.
Информатика и
программирование.
Практикум**

Доцент Петрович

Беспилотники и нейросети.

Доцент Петрович

**Беспилотники и нейросети.
Информатика и
программирование. Практикум**

«Автор»

2026

Петрович Д.

Беспилотники и нейросети. Информатика и программирование.
Практикум / Д. Петрович — «Автор», 2026 — (Беспилотники и
нейросети.)

Книга "Беспилотники и нейросети. Информатика и программирование. Практикум" продолжает серию моих книг, посвященных изучению в школе и техникуме вопросов разработки, устройства и применения беспилотных летательных аппаратов (БПЛА). Но, в отличие от первой моей книги "Беспилотники и нейросети. Физика. Сборник задач", в этой книге в центре внимания находятся информационные аспекты БПЛА, и на практических занятиях и лабораторных работах рассматривается весь цикл вопросов разработки и создания программного обеспечения таких и аналогичных систем и комплексов. Книга может быть интересна учителям, преподавателям, студентам и старшеклассникам.

© Петрович Д., 2026

© Автор, 2026

Содержание

Информационные аспекты изучения беспилотных летательных аппаратов(БПЛА) в школе и техникуме	5
Языки программирования.и нейросети. Решение задач.	8
1.Работа с графическим интерфейсом языка программирования	9
2. Создание алгоритмов и отладка программ линейной структуры	12
3. Создание алгоритмов и отладка программ с использованием логических высказываний	21
4.Решение задач с помощью условных операторов на языке Паскаль	29
5.Решение задач по геометрии с помощью Систем Программирования на языках Паскаль и Питон	37
6.Операторы цикла	48
Конец ознакомительного фрагмента.	50

Беспилотники и нейросети. Информатика и программирование. Практикум

Информационные аспекты изучения беспилотных летательных аппаратов(БПЛА) в школе и техникуме

Изучение беспилотных летательных аппаратов (БПЛА) является перспективной и востребованной областью современной образовательной программы, особенно актуальной в условиях развития современных технологий.

В школах и техникумах изучение БПЛА способствует формированию у учащихся современных знаний и компетенций, необходимых для успешного участия в высокотехнологичных отраслях промышленности, науки и военного дела.

Информационные аспекты

1. Учебно – методическое и программное обеспечение.

1)Учебники и пособия:

специализированные учебники и методические пособия, содержащие теоретический материал по основам аэродинамики, теории полета, принципам работы различных типов БПЛА, элементам конструкции и управления.

2)Электронные ресурсы:

интерактивные онлайн - курсы, видеоматериалы, презентации, вебинары, симуляторы полетов и тренажеры, позволяющие учащимся получить практические навыки дистанционного управления аппаратами.

3)Программное обеспечение:

программное обеспечение для моделирования полетов, проектирования и анализа характеристик БПЛА, позволяющее студентам разрабатывать собственные проекты и проводить виртуальные эксперименты.

2. Практическая подготовка и лабораторные занятия

1)Лабораторные стенды и оборудование:

наборы конструкторов и учебных моделей БПЛА, лаборатории робототехники и авиамоделизма, оснащенные современными средствами диагностики и контроля.

2)Практическое обучение:

проведение практических занятий по сборке, настройке и запуску БПЛА, отработка навыков пилотирования и выполнения полетных заданий, участие в соревнованиях и конкурсах.

3)Проектная деятельность:

выполнение индивидуальных и групповых проектов по созданию собственных моделей БПЛА, разработка программного обеспечения для управления и обработки данных, полученных с помощью дронов.

3. Интеграция междисциплинарных знаний

Математика и физика[2]:

изучение законов механики, аэродинамики, динамики полета, принципов навигации и ориентации в пространстве.

Информатика и программирование[3,4,5]:

1)освоение алгоритмов управления, программирования микроконтроллеров, разработки программного обеспечения для автономных систем управления.

2)робототехника и электроника[1]:

3) основы электроники, конструирования и сборки электронных компонентов, используемых в системах управления БПЛА.

4) геодезия и картография[1]:

5) использование данных GPS/ГЛОНАСС, обработка спутниковых снимков, создание цифровых карт и планов местности.

4. Профессиональная ориентация и карьерные перспективы

Профессиональные компетенции[1]:

1) формирование профессиональных навыков, востребованных в авиационной отрасли, при фото и видеосъемке, сельском хозяйстве, строительстве, спасательных операциях, экологическом мониторинге, военном деле и других сферах.

2) Карьерные возможности и рост:

3) получение квалификаций и сертификатов, подтверждающих уровень подготовки специалистов по эксплуатации и обслуживанию БПЛА, возможность дальнейшего трудоустройства в компаниях, занимающихся разработкой, производством и эксплуатацией беспилотников.

5. Социальная значимость и этика[1]

1. Безопасность полетов: изучение правил безопасности, законодательства, регулирующего эксплуатацию БПЛА, предотвращение аварийных ситуаций и минимизация рисков.

2. Этика и ответственность:

3. Обсуждение вопросов ответственности перед обществом, конфиденциальности личных данных, охраны окружающей среды и культурного наследия при использовании БПЛА.

Таким образом, информационные аспекты изучения беспилотных летательных аппаратов включают комплексную образовательную программу, охватывающую теоретические знания, практическую подготовку, интеграцию междисциплинарных дисциплин, развитие профессиональной компетентности и осознание социальной значимости и этической стороны использования данной технологии.

Вопросы информатики и программирования сегодня занимают центральное место в процессе изучения беспилотных летательных аппаратов (БПЛА) в образовательных учреждениях — школах и техникумах. Эти дисциплины обеспечивают фундаментальную основу для понимания принципов функционирования, управления и программирования бортового оборудования и наземных станций управления БПЛА.

Курс практических и лабораторных занятий «Языки программирования и нейросети. Решение задач» направлен на формирование профессиональных компетенций студентов и школьников в области разработки программного обеспечения и решения прикладных задач средствами современных языков программирования (на примере Python и Pascal). Программа курса охватывает основные этапы разработки программного продукта: создание алгоритма, написание программы, её тестирование и отладка, а также применение полученных знаний для реализации типовых задач, включая обработку одномерных и двумерных массивов, решение геометрических задач и программирование циклических структур.

Целью данного курса является развитие у обучающихся навыков практического программирования, анализа и проектирования алгоритмов, а также формирования представления о принципах работы и возможностях современных нейронных сетей и методов машинного обучения.

Курс включает в себя следующие разделы:

1. Работа с графическим интерфейсом языка программирования – изучение основ взаимодействия с интерфейсами систем программирования и создание простых приложений с использованием визуальных компонентов.

2. Создание алгоритмов и отладка программ линейной структуры – освоение базовых принципов построения последовательных алгоритмов и разработка первых самостоятельных программ.

3. Использование логических высказываний в программах – закрепление знаний о работе с условными операторами и реализация программ с ветвлениями.

3. Создание алгоритмов и отладка программ с использованием логических высказываний.

4. Создание алгоритмов и отладка программ с использованием логических операций.

5. Решение задач по геометрии с помощью программирования – обучение применению математических знаний и навыков программирования для автоматизации вычислений и решений геометрических задач.

6. Программирование циклов и обработка массивов – ознакомление с основными видами циклических конструкций и методами обработки числовых массивов различной размерности.

7. Отладка и тестирование программ на примере решения реальных задач – получение опыта выявления ошибок и исправления дефектов в коде, а также проверка правильности выполнения программ на конкретных примерах.

8. Практическое программирование нейросетей и машинного обучения – знакомство с базовыми концепциями нейросетевых технологий и их применением для решения прикладных задач.

Методы и формы обучения

В рамках курса предусмотрены практические занятия, лабораторные работы и индивидуальные задания, направленные на закрепление теоретических знаний и приобретение необходимых практических навыков. Практические занятия проводятся в компьютерных классах, где студенты и школьники работают над решением предложенных задач и получают обратную связь от преподавателя.

Итоговый результат освоения курса

По завершении курса студент (школьник) должен уметь разрабатывать эффективные алгоритмы, реализовывать их на языках программирования высокого уровня, проводить тестирование и отладку созданных программ, а также понимать базовые принципы работы нейросетевых моделей и использовать их возможности для решения прикладных задач.

Таким образом, курс способствует формированию у студентов фундаментальной базы знаний и навыков, необходимых для успешной профессиональной деятельности в сфере информационных технологий и программирования.

Языки программирования.и нейросети. Решение задач.

1. Работа с графическим интерфейсом языка программирования

Сегодня одним из центральных вопросов при изучении языков программирования является вопрос выбора системы программирования того или другого языка. Этот выбор затем определяет весь процесс обучения.

Давайте такой вопрос зададим нейросети GPTutor (ИИЗ):



Николай

Какие системы программирования на языке Паскаль популярны сегодня? Дайте им краткую характеристику.



GPTutor

Сегодня в основном используются две популярные системы программирования на языке Паскаль:

1. Free Pascal (FPC): Это бесплатная и открытая среда разработки, которая поддерживает множество платформ, включая Windows, Linux, macOS и др. Free Pascal обладает обширной библиотекой функций, хорошей совместимостью с Turbo Pascal и Delphi, что делает её популярной для разработки прикладного ПО.
2. Lazarus: Это среда разработки с открытым исходным кодом для создания графических приложений на базе свободного компилятора Free Pascal. Lazarus поддерживает создание кроссплатформенных приложений, работает на различных операционных системах и имеет удобный визуальный интерфейс.

Обе системы позволяют разработчикам использовать язык Паскаль для создания мощных и современных программных решений.

О системе программирования PascalABC.NET ни слова. А она, например, в колледже при политехническом университете Петра Великого являлась основной. Поэтому с моей стороны следует естественный дополнительный вопрос:



Николай
А Pascal ABC ?



GPTutor

[PascalABC.NET](#) (также известная как "Pascal ABC") - это среда разработки и язык программирования, разработанные для обучения и преподавания программирования. Основной упор [PascalABC.NET](#) делает на простоту и понятность для новичков, предоставляя интуитивно понятный интерфейс и набор инструментов для изучения основ программирования.

Язык программирования [PascalABC.NET](#) включает множество возможностей, направленных на обучение и облегчение изучения основ программирования, включая широкую поддержку графики, удобную систему ввода-вывода, интегрированные инструменты отладки и многие другие полезные функции.

[PascalABC.NET](#) остается популярным инструментом для введения студентов и начинающих разработчиков в мир программирования, обеспечивая простоту и удобство в использовании при изучении основных концепций программирования.

Должен сказать, что я не ожидал получить столь полные и аргументированные ответы на эти вопросы.

Узловые вопросы программирования нейросети, судя по GPTutor представляют неплохо.

Работа с графическим интерфейсом языка программирования Паскаль включает в себя три базовых компонента:

1.1.Работу со Справочником по системе ABC Pascal

1. В окне системы **ABC Pascal** открыть меню *Помощь* (рис.1).
2. В открывшемся меню выбрать пункт **СПРАВКА**.
3. В открывшемся окне **СПРАВКА** (рис.1) работать с его содержанием.

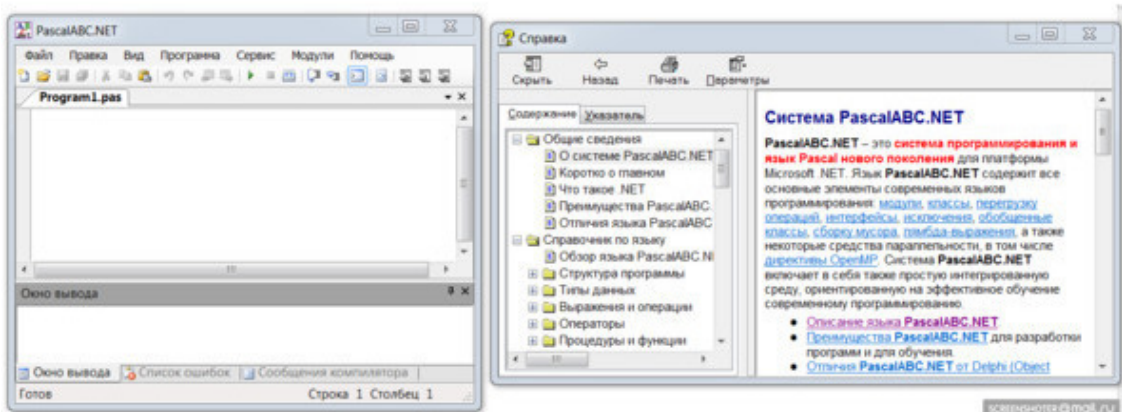


Рис.1. Окна систем программирования Pascal ABC и Справки

1.2. Работа с сайтом «Уроки Pascal ABC для начинающих» (Ссылка на сайт в конце книги)



Профессиональная среда разработки для создания программ и приложений любого уровня сложности. Сочетает в себе классическую простоту Паскаля и все возможности современной среды разработки .NET, которую используют профессиональные разработчики по всему миру.

1.3. Изучение презентации и видео «Паскаль. Обзор»

В основе данных презентации и видео лежит курс обучения программированию на языке Паскаль на базе учебного пособия «Основы алгоритмизации и программирование на языке Паскаль», с 4х главным содержанием.

С презентацией на тему «Паскаль _обзор» для 8 и 9 класса и видео на эту же тему можно познакомиться, воспользовавшись ссылками в конце книги.

2. Создание алгоритмов и отладка программ линейной структуры

Вот теперь перейдем к решению первых задач по программированию на языке Паскаль. Начнем делать первые шаги и рассмотрим первые 3 простейшие задачи, имеющие линейную структуру алгоритмов их решения. По каждой из задач представлено 2 вида решений:

1.Традиционный (обычный) способ

Программа записана на языке Паскаль, ее реализация оформлена в системе программирования **Pascal ABC.NET** и приведены пояснения и комментарии.

2.Решение задачи с помощью нейросети

Программа записана на языке Паскаль и приведены пояснения и комментарии

2.1.Решение задач

Задача 1.

Дано: сторона квадрата a . Найти его периметр $P = 4 \cdot a$

1.Традиционное решение (см. рис.2).

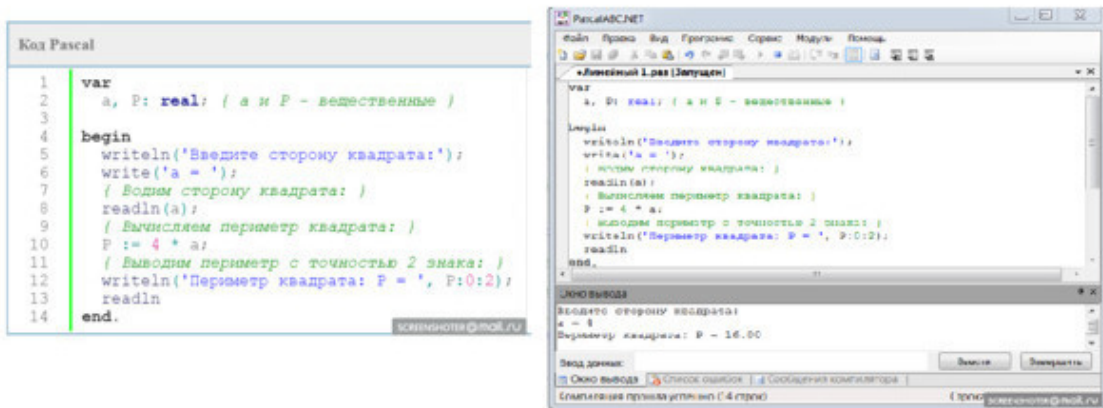


Рис.2.Программа на языке Паскаль и ее реализация в системе программирования Pascal ABC.NET.

Смотрим основные моменты по строкам программы:

- 1) В первой строке у нас идет описание переменных, и на это указывает служебное слово var.
- 2) Описываем переменные a , P как вещественные (real – вещественный тип данных).
- 4) Служебное слово begin указывает на начало программы. После него идет описание операторов, которые программа будет выполнять один за другим.
- 5) Вводим длину стороны квадрата. При этом в строках 5 и 6 выводим на экран информацию о запросе ввода.
- 6) Вычисляем периметр квадрата по формуле $P = 4 \cdot a$.
- 7) Выводим результат, выделяя для результата 2 знака после запятой.
- 8) readln означает ожидание программы до нажатия клавиши Enter – после этого программа продолжит свое выполнение.
- 9) Наконец, служебное слово end с точкой в конце означает завершение программы.

2.Решение задачи с помощью нейросети (рис.3 – 11):

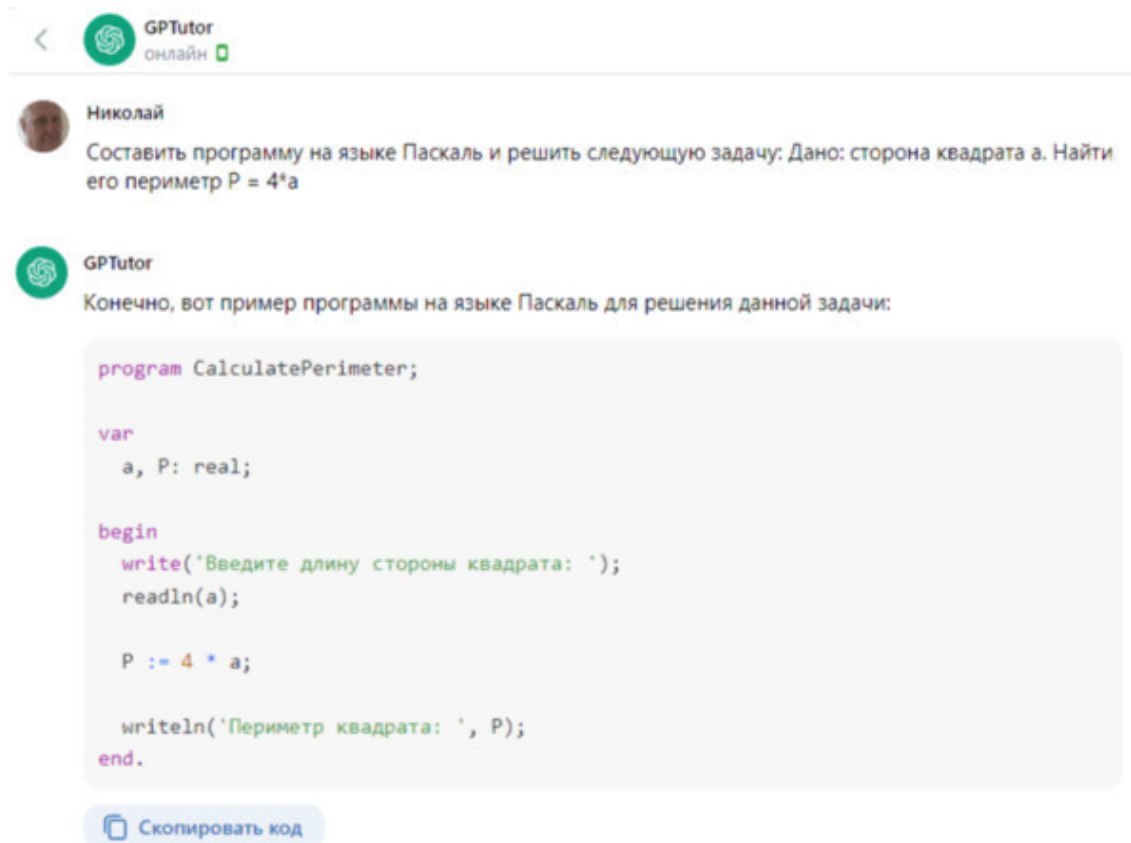


Рис.3.

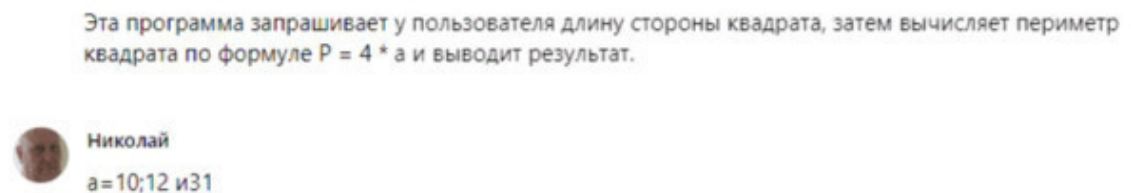


Рис.4.

Я пытаюсь спровоцировать нейросеть на решение этой задачи и ввожу исходные данные $a = 10$; $a = 12$; $a = 31$.

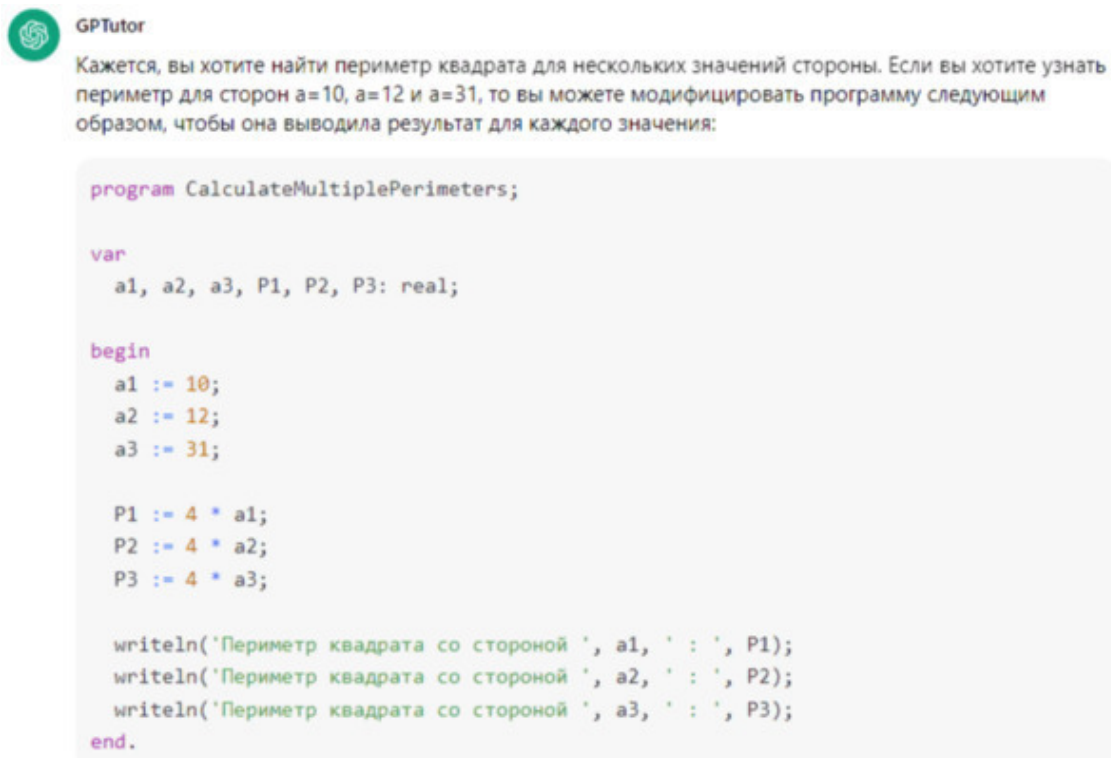


Рис.5.

Нейросеть восприняла мой запрос, как необходимость модифицировать коды программы.

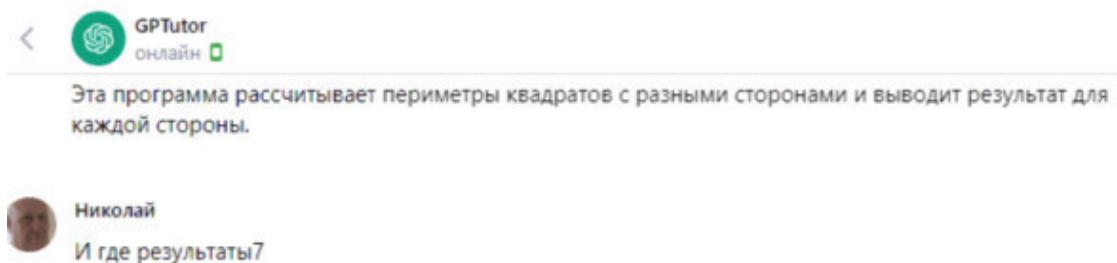



Рис.6.

 **GPTutor**
Извините за путаницу. Программа выше не предусматривает вывод в консоль. Вот обновленная версия программы, где результаты выводятся в консоль:

```
program CalculateAndShowPerimeters;

var
  a1, a2, a3, P1, P2, P3: real;

begin
  a1 := 10;
  a2 := 12;
  a3 := 31;

  P1 := 4 * a1;
  P2 := 4 * a2;
  P3 := 4 * a3;

  writeln('Периметр квадрата со стороной ', a1, ' : ', P1);
  writeln('Периметр квадрата со стороной ', a2, ' : ', P2);
  writeln('Периметр квадрата со стороной ', a3, ' : ', P3);

  readln; // Чтобы окно консоли не закрывалось сразу
end.
```

Рис.7.

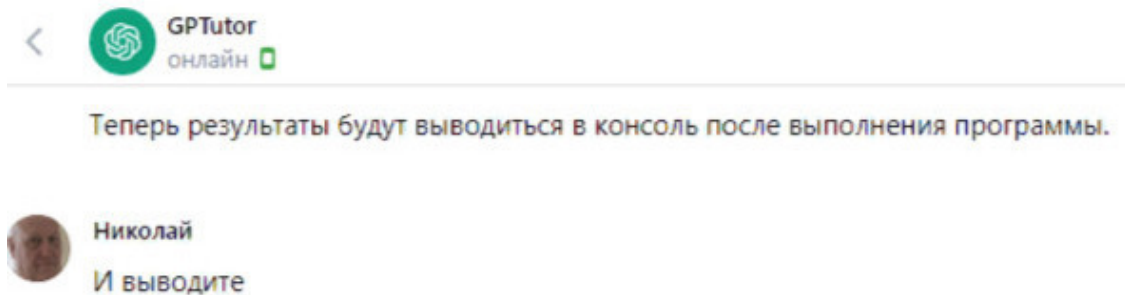


Рис.8.

Я все время требую от **GPTutor** невозможного: нельзя решить эту задачу на ПК, без предварительной загрузки и отладки программы этой задачи в системе **Pascal ABC.NET**. А нейросеть **GPTutor** при этом еще оправдывается и «чувствует себя виноватой». И предлагает новые варианты модификации программы этой задачи.



GPTutor

Извините за путаницу. Вот исправленная версия программы, которая выводит результаты в консоль:

```
program CalculateAndShowPerimeters;

var
  a1, a2, a3, P1, P2, P3: real;

begin
  a1 := 10;
  a2 := 12;
  a3 := 31;

  P1 := 4 * a1;
  P2 := 4 * a2;
  P3 := 4 * a3;

  writeln('Периметр квадрата со стороной ', a1, ' : ', P1);
  writeln('Периметр квадрата со стороной ', a2, ' : ', P2);
  writeln('Периметр квадрата со стороной ', a3, ' : ', P3);

  readln; // Чтобы окно консоли не закрывалось сразу
end.
```

Рис.9.

Теперь после выполнения программы вы увидите результаты в консоли.



Николай

a=67

Рис.10.

Я продолжаю вводить новые исходные данные, с большим интересом ожидая реакции GPTutor.



GPTutor

Если вы хотите изменить значение стороны квадрата на 67, вы можете просто изменить переменную `a1` в программе на 67, как показано ниже:

```
program CalculateAndShowPerimeters;

var
  a1, P1: real;

begin
  a1 := 67;

  P1 := 4 * a1;

  writeln('Периметр квадрата со стороной ', a1, ' : ', P1);

  readln; // Чтобы окно консоли не закрывалось сразу
end.
```

Скопировать код

После выполнения этой программы в консоли будет отображено значение периметра для стороны квадрата равной 67.

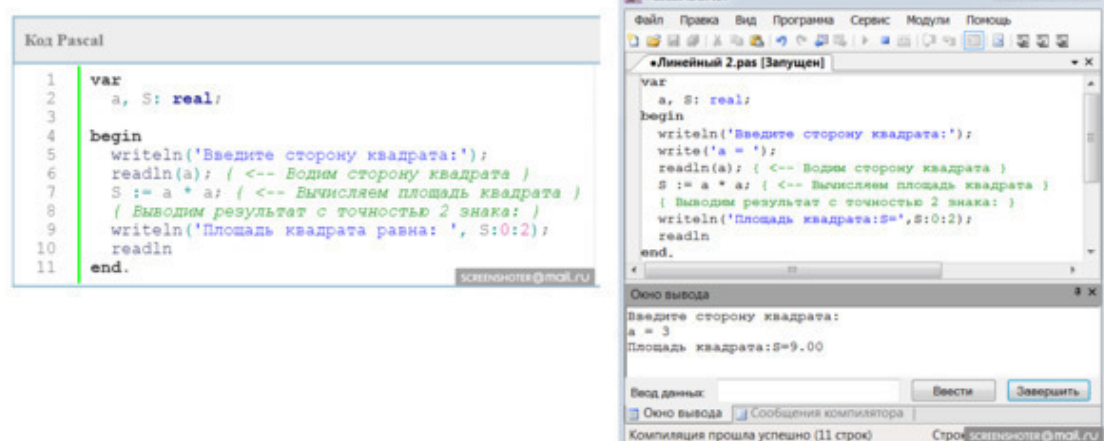
Рис.11.

Не знаю, как вам, а мне все действия нейросети очень понравились. На основе данной задачи нейросеть **GPTutor** провела мастер – класс по программированию на языке Паскаль.

Задача 2.

Дано: сторона квадрата a . Найти его площадь $S = a^2$.

1.Традиционное решение (рис.12)



- 2) Вычисляем площадь квадрата $S = a^2$ непосредственным умножением. Для вычисления квадрата числа мы могли бы использовать функцию $\text{sq}(x)$.
- 3) Выводим значение площади с точностью до двух знаков после запятой.

2.Решение задачи с помощью нейросети (рис.13 и 14)

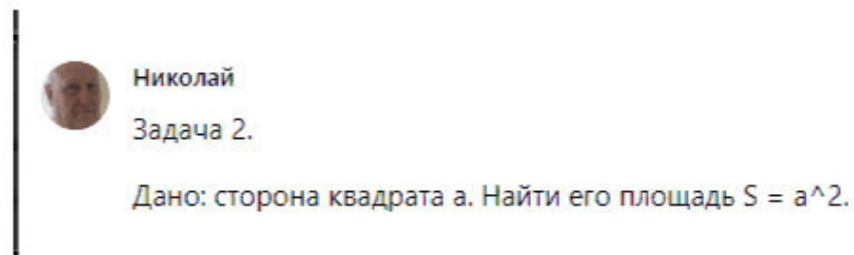


Рис.13.

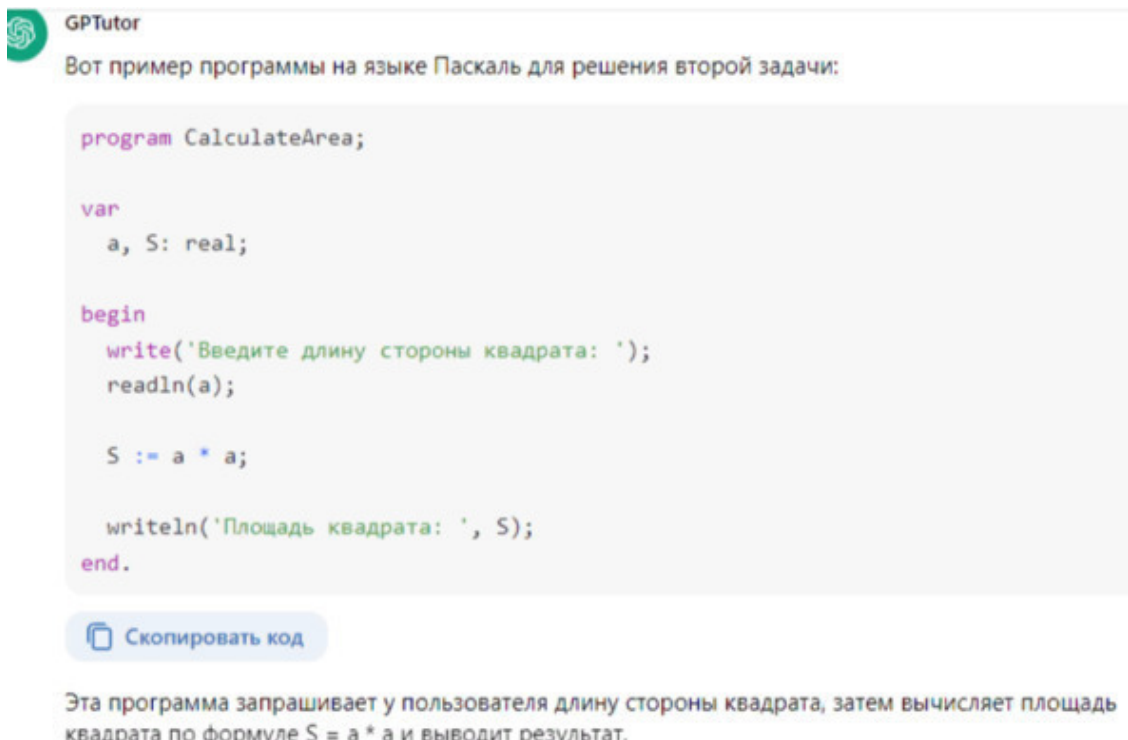


Рис.14.

Задача 3.

Дан диаметр окружности d . Найти ее длину $L = \pi * d$. В качестве значения π использовать 3,14.

1.Традиционное решение (см. рис.15).

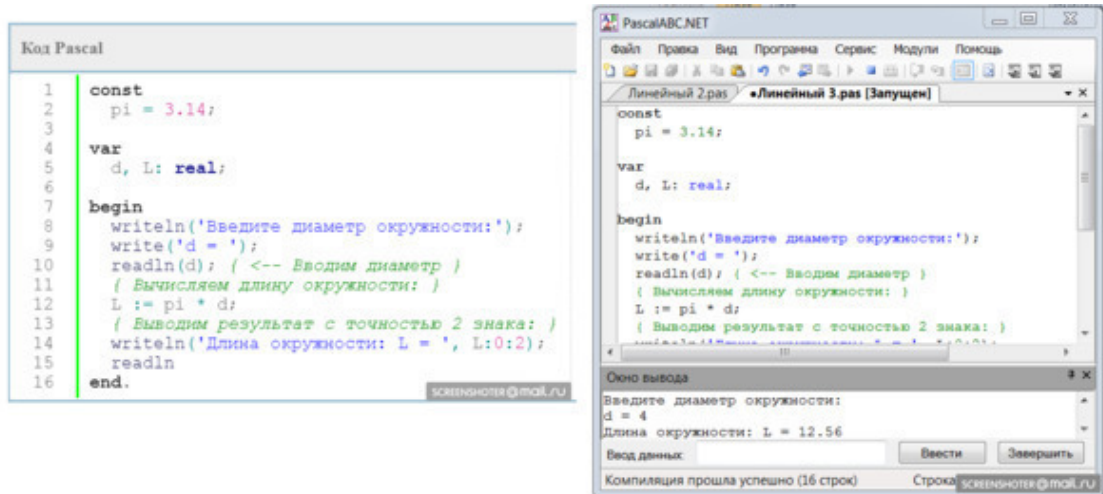


Рис.15. Программа на языке Паскаль и ее реализация в системе программирования **Pascal ABC.NET**

Здесь нужно сделать небольшой комментарий относительно числа «пи». Как упоминалось в начале, в паскале есть встроенная переменная для обозначения числа «пи»: $\pi = 3.141592653589793$. Поэтому, если бы мы в разделе описания констант не записали $\pi = 3.14$ (строки 1—2 после служебного слова const), то в нашей программе было бы использовано встроенное значение числа «пи» большей точности.

2.Решение задачи с помощью нейросети (рис.16 —17):

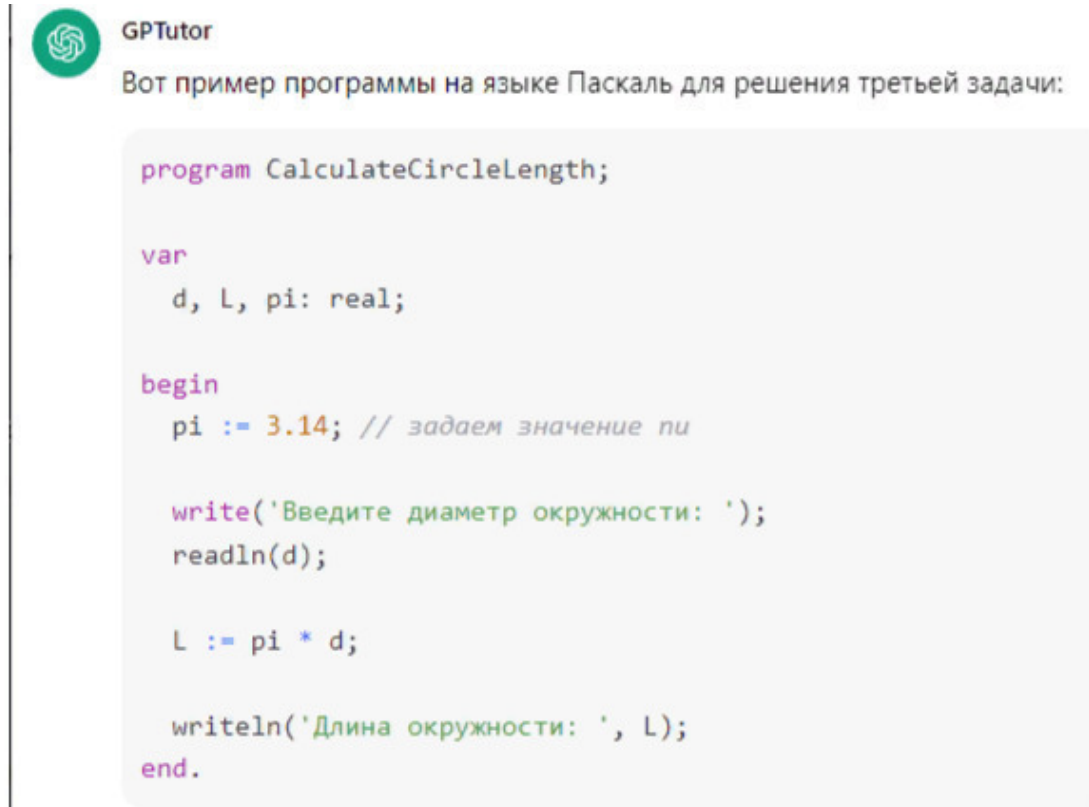


Николай

Задача 3.

Дан диаметр окружности d . Найти ее длину $L = \pi \cdot d$. В качестве значения π использовать 3,14

Рис.16.



The screenshot shows the GPTutor logo (a green circle with a white knot) and the text "GPTutor". Below it, the text reads "Вот пример программы на языке Паскаль для решения третьей задачи:". The code is displayed in a light gray box with syntax highlighting. The code is as follows:

```
program CalculateCircleLength;

var
  d, L, pi: real;

begin
  pi := 3.14; // задаем значение pi

  write('Введите диаметр окружности: ');
  readln(d);

  L := pi * d;

  writeln('Длина окружности: ', L);
end.
```

Рис.17.

Реализация решения этих трех задач в программах на языке Паскаль нейросетью **GPTutor** показывает ее широкие и интересные возможности не только по решению данных задач, но и по обучению и подготовке специалистов на вариантах модификации программ этих задач.

3. Создание алгоритмов и отладка программ с использованием логических высказываний

3.1. Постановка задач

Задача 1

Дано целое положительное число. Проверить истинность высказывания: «Данное число является четным двузначным».

Задача 2

Дано целое положительное число. Проверить истинность высказывания: «Данное число является нечетным трехзначным».

Задача 3

Дано трехзначное число. Проверить истинность высказывания: «Все цифры данного числа различны».

Задача 4

Дано трехзначное число. Проверить истинность высказывания: «Цифры данного числа образуют возрастающую последовательность».

Задача 5

Даны три целых числа: A, B, C . Проверить истинность высказывания: «Каждое из чисел A, B, C положительное».

Задача 6

Даны три целых числа: A, B, C . Проверить истинность высказывания: «Хотя бы одно из чисел A, B, C положительное».

Задача 7

Даны три целых числа: A, B, C . Проверить истинность высказывания: «Ровно одно из чисел A, B, C положительное».

Задача 8

Проверить истинность высказывания: «Среди трех данных целых чисел есть хотя бы одна пара совпадающих».

**3.2.Решение задач с использованием логических высказываний и операций на языке Паскаль. Результаты
РЕШЕНИЕ:**

Задача 1

```
const a=98;  
var  
    flag:boolean;  
begin  
    flag:=false;  
    if (a div 100 =0) and (a div 10 <>0) and (a mod 2=0) then  
        flag:=true;  
    writeln(flag);  
end.
```

Окно вывода

True

Задача 2

```
const a=888;  
var  
    f_n, s_n, t_n: integer;  
    flag:boolean;  
begin  
    flag:=false;  
    if (a div 100 <>0) and(a div 10 <>0) and (a div 1000=0) and  
        flag:=true;  
    writeln(flag);  
end.
```

Окно вывода

False

Задача 3

```
const a=348;
var
  f_n, s_n, t_n: integer;
  flag:boolean;
begin
  flag:=false;
  f_n:=a div 100;
  s_n:=(a mod 100)div 10;
  t_n:=a mod 10;
  if (f_n<>s_n) and (f_n<>t_n) and (s_n<>t_n) then
    flag:=true;
  writeln(flag);
end.
```

Окно вывода
True

Задача 4

```
const a=318;
var
  f_n, s_n, t_n: integer;
  flag:boolean;
begin
  flag:=false;
  f_n:=a div 100;
  s_n:=((a mod 100)div 10);
  t_n:=a mod 10;
  if (f_n<s_n) and (s_n<t_n) then flag:=true;
  writeln(flag);
end.
```

Окно вывода
False

Задача 5

```
write('A = ');
readln(A);
write('B = ');
readln(B);
write('C = ');
readln(C);
writeln;
{ Поскольку условия A>0, B>0 и C>0 должны выполняться
одновременно, то объединяем их оператором and: }
writeln('Каждое из чисел положительное? ',
(A > 0)and(B > 0)and(C > 0)); { <-- Выводим значение }
readln
end.
```

Окно вывода

```
A = 3
B = 4
C = 12
Каждое из чисел положительное? True
```

Задача 6

```
begin
writeln('Введите целые числа A, B, C:');
readln(A, B, C); { <-- Вводим три целые числа A, B, C }
writeln;
{ Поскольку среди условий A>0, B>0 и C>0 должно выполняться
хотя бы одно, то объединяем их оператором or (логическое
"или"). Если получим результат True, то среди введенных чисел
есть положительное, в противном случае мы получим False: }
writeln('Хотя бы одно из чисел A, B, C положительное? ',
(A > 0)or(B > 0)or(C > 0));
readln
end.
```

Окно вывода

```
-2
3
-8
Хотя бы одно из чисел A, B, C положительное? True
```

Задача 7

```
readln(A, B, C); { <-- Вводим три целые числа A, B, C }
{ Составляем три выражения, каждое из которых означает,
  что только одно из чисел положительное: }
f := (A > 0)and(B <= 0)and(C <= 0); { <== «только A > 0» }
g := (B > 0)and(A <= 0)and(C <= 0); { <== «только B > 0» }
h := (C > 0)and(A <= 0)and(B <= 0); { <== «только C > 0» }
writeln:
{ Поскольку среди условий f, g и h может выполняться только
  одно, то объединяем их оператором xor ("исключающее или"): }
writeln('Ровно одно из чисел положительное? ', f xor g xor h);
readln
end.
```

Окно вывода

```
2
-3
-11
Ровно одно из чисел положительное? True
```

Задача 8

```
A, B, C: integer;

begin
  writeln('Введите три целые числа:');
  readln(A, B, C);
  writeln:
  { Поскольку нужно проверить совпадение хотя бы одной
    пары чисел, то для проверки трех пар используем
    оператор or (логическое "или"): }
  writeln(' ', (A = B) or (B = C) or (C = A));
  readln
end.
```

Окно вывода

```
Введите три целые числа:
2
2
7
True
```

№ задачи	Содержание	Результат
1.	<i>Дано целое положительное число. Проверить истинность высказывания: «Данное число является четным двузначным».</i>	True
2.	<i>Дано целое положительное число. Проверить истинность высказывания: «Данное число является нечетным трехзначным».</i>	False
3.		True
4.	<i>Дано трехзначное число. Проверить истинность высказывания: «Цифры данного числа образуют возрастающую последовательность».</i>	False

№ задачи	Содержание	Результат
5.	<p><i>Даны три целых числа: A, B, C. Проверить истинность высказывания: «Каждое из чисел A, B, C положительное».</i></p>	<pre>A = 3 B = 4 C = 12 Каждое из чисел положительное? True</pre>
6.	<p><i>Даны три целых числа: A, B, C. Проверить истинность высказывания: «Хотя бы одно из чисел A, B, C положительное».</i></p>	<pre>-2 3 -8 Хотя бы одно из чисел A, B, C положительное? True</pre>
7.	<p><i>Даны три целых числа: A, B, C. Проверить истинность высказывания: «Ровно одно из чисел A, B, C положительное».</i></p>	<pre>2 -3 -11 Ровно одно из чисел положительное? True</pre>
8.	<p><i>Проверить истинность высказывания: «Среди трех данных целых чисел есть хотя бы одна пара совпадающих».</i></p>	<pre>Введите три целые числа: 2 2 7 True</pre>

4.Решение задач с помощью условных операторов на языке Паскаль

4.1.Постановка задач

Задача 1

Дано целое число N. Если оно положительное, то прибавить к нему 1; если отрицательно вычесть из него 2. Вывести полученное число.

Задача 2

Ввести число. Если оно неотрицательно, вычесть из него 10, в противном случае прибавить к нему 10.

Задача 3

Ввести два числа. Вычесть из большего меньшее.

Задача 4

Даны два числа. Вывести большее из них.

Задача 5

Даны две переменные **A**, **B**. Поместить в **A** меньшее из значений, а в **B** — большее. Вывести новые значения переменных **A** и **B**.

Задача 6

Напечатать ряд из повторяющихся чисел 20 в виде:
20 20 20 20 20 20.

Задача 7

Напечатать столбиком все целые числа от 20 до 35 и вычислить корни этих чисел.

Задача 8

Вычислить сумму ряда $S = 1+1.5+2+2.5+3+3.5+\dots+30$.

4.2.Решение задач с помощью условных операторов на языке Паскаль

Задача 1

```
Program Zad_1;  
var n:integer;  
begin  
write ('введите целое число n=');  
readln(n);  
if n>0 then n:=n+1 else n:=n-2;  
writeln ('n=',n);  
end.
```

Окно вывода

```
введите целое число n=10  
n=11
```

Окно вывода

```
введите целое число n=-10  
n=-12
```

Задача 2

```
Var a:integer;  
Begin  
writeln ('Введите значение переменной a');  
Readln (a);  
If a>0 then a:= a -10 else a:= a+10;  
Writeln ('Переменная a = ',a);  
End.
```

Окно вывода

```
Введите значение переменной a  
20  
Переменная a = 10
```

Задача 3

```
Var a,b,d:integer;  
Begin  
writeln ('Введите значение переменных a и b');  
Readln (a,b);  
If a>b then d:=a-b else d:=b-a;  
Writeln ('Переменная d = ',d);  
End.
```

Окно вывода

```
Введите значение переменных a и b  
14 11  
Переменная d = 3
```

Задача 4

```
Program a_max_b;  
var a,b,max:real;  
begin  
writeln ('введите числа a и b');  
readln(a,b);  
writeln ('a=',a,'b=',b);  
max:=b;  
if a>b then max:=a;  
writeln ('max=',max);  
end.
```

Окно вывода

```
введите числа a и b  
10 15  
a=10b=15  
max=15
```

Ввод чисел осуществляем через пробел

Задача 5

```
program aib;  
var a,b,v:real;  
begin  
  writeln ('введите числа а и b');  
  readln(a,b);  
  writeln ('a=',a,' b=',b);  
  if a>b then begin v:=a; a:=b; b:=v; end;  
  writeln ('a=',a,' b=',b);  
end.
```

Окно вывода

```
введите числа а и b  
5 4  
a=5 b=4  
a=4 b=5
```

Задача 6

```
var i: integer;  
begin  
  for i:=1 to 6 do write(20,' ')  
end.
```

Окно вывода

```
20 20 20 20 20 20 |
```

Задача 7

```
var i: integer;
begin
for I:=20 to 35 do writeln(I, ' ',sqrt(i):0:1);
end.
```

Окно вывода

```
20 4.5
21 4.6
22 4.7
23 4.8
24 4.9
25 5.0
26 5.1
27 5.2
28 5.3
29 5.4
30 5.5
31 5.6
32 5.7
33 5.7
34 5.8
35 5.9
```

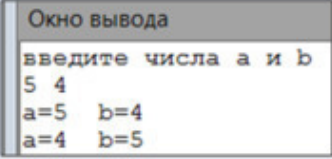
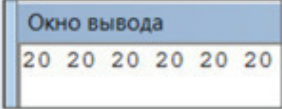
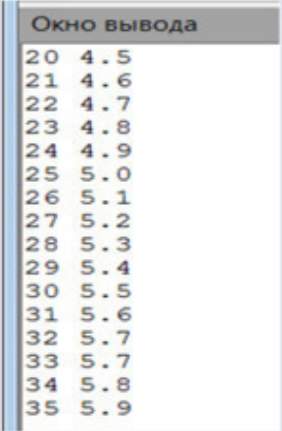
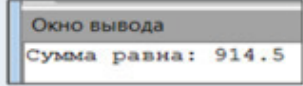
Задача 8

```
var
sum:real;
n:real;
BEGIN
sum:=0;
n:=1;
while n <=30 do
begin
sum:=sum+n;
n:=n+0.5;
end;
writeln('Сумма равна: ',sum);
END.
```

Окно вывода

```
Сумма равна: 914.5
```

№ задачи	Содержание	Результат
1.	Дано целое число N. Если оно положительное, то прибавить к нему 1; если отрицательно вычесть из него 2. Вывести полученное число.	<div data-bbox="916 304 1294 412" style="border: 1px solid black; padding: 2px;"> <p>Окно вывода</p> <p>введите целое число n=10 n=11</p> </div> <div data-bbox="916 421 1294 528" style="border: 1px solid black; padding: 2px;"> <p>Окно вывода</p> <p>введите целое число n=-10 n=-12</p> </div>
2.	Ввести число. Если оно неотрицательно, вычесть из него 10, в противном случае прибавить к нему 10.	<div data-bbox="916 577 1294 696" style="border: 1px solid black; padding: 2px;"> <p>Окно вывода</p> <p>Введите значение переменной a 20 Переменная a = 10</p> </div>
3.	Ввести два числа. Вычесть из большего меньшее.	<div data-bbox="916 801 1294 920" style="border: 1px solid black; padding: 2px;"> <p>Окно вывода</p> <p>Введите значение переменных a и b 14 11 Переменная d = 3</p> </div>
4.	Даны два числа. Вывести большее из них.	<div data-bbox="983 976 1230 1117" style="border: 1px solid black; padding: 2px;"> <p>Окно вывода</p> <p>введите числа a и b 10 15 a=10b=15 max=15</p> </div>

№ задачи	Содержание	Результат
5.	Даны две переменные A, B . Поместить в A меньшее из значений, а в B — большее. Вывести новые значения переменных A и B .	 <pre> Окно вывода введите числа a и b 5 4 a=5 b=4 a=4 b=5 </pre>
6.	Напечатать ряд из повторяющихся чисел 20 в виде: 20 20 20 20 20 20 .	 <pre> Окно вывода 20 20 20 20 20 20 </pre>
7.	Напечатать столбиком все целые числа от 20 до 35 и вычислить корни этих чисел.	 <pre> Окно вывода 20 4.5 21 4.6 22 4.7 23 4.8 24 4.9 25 5.0 26 5.1 27 5.2 28 5.3 29 5.4 30 5.5 31 5.6 32 5.7 33 5.7 34 5.8 35 5.9 </pre>
8.	Вычислить сумму ряда $S = 1+1.5+2+2.5+3+3.5+\dots+ 30$.	 <pre> Окно вывода Сумма равна: 914.5 </pre>

5. Решение задач по геометрии с помощью Систем Программирования на языках Паскаль и Питон

Сегодня в школах и техникумах на занятиях по изучению систем программирования (СП) все чаще СП на языке Паскаль заменяют СП на языке Питон. В связи с этим, в этой главе на примере 4х задач по геометрии показано, как с этими задачами справляется каждая из этих СП.

5.1. Постановка задач

Задача 1

Даны длины ребер a, b, c прямоугольного параллелепипеда. Найти его объем $V = abc$ и площадь поверхности $S = 2(ab + bc + ac)$.

Задача 2

Дана длина ребра куба. Найти площадь грани ($S_1 = a^2$), площадь полной поверхности ($S_2 = 6a^2$) и объем куба ($V = a^3$).

Задача 3

Составьте программу нахождения площади равнобедренной трапеции через стороны a, b, c .

Задача 4

Составить программу, которая по трем введенным вами числам определит, могут ли эти числа быть длинами сторон треугольника.

5.2. Решение задач

5.2.1. Решение задач с помощью системы программирования на языке «Паскаль». Результаты

Задача 1

```
var a,b,c: integer;
var V,S: real;
begin
a:=6;
b:=3;
c:=9;
V:=a*b*c;
S:=2*(a*b+b*c+a*c);
writeln (V);
writeln (S);
end.
```

Окно вывода

```
162
198
```

Задача 2

```
Program Kub;
var a: integer;
S1, S2, V: real;
begin
readln (a);
S1:= sqr(a);
S2:= 6*sqr(a);
V:=sqr(a)*a;
writeln ('Площадь грани куба = ', S1);
writeln ('Площадь полной поверхности куба = ', S2);
writeln ('Объем куба = ', V);
end.
```

Окно вывода

```
12
Площадь грани куба = 144
Площадь полной поверхности куба = 864
Объем куба = 1728
```

Задача 3

```
var a, b, c: integer;
S: real;
begin
  readln (a,b,c);
  S:=(a+b)/2*(sqrt(sqr(c)-sqr(a-b)/4));
  writeln ('формула S = ', S)
end.
```

Окно вывода

```
3 5 7
формула S = 27.712812921102
```

Задача 4

```
Var a,b,c: integer;
Begin
  writeln ('Введите значение переменных a, b, c');
  Readln (a,b,c);
  If a>=b+c then Writeln ('Нет')
  Else if b>=a+c then Writeln ('Нет')
  Else if c>=a+b then Writeln ('Нет')
  Else Writeln ('Да');
End.
```

Окно вывода

```
Введите значение переменных a, b, c
5 7 9
Да
```

5.2.2.Решение задач с помощью системы программирования на языке «Питон». Результаты

Задача 1

```
python Copy code  
  
# функция для вычисления объема и поверхности прямоугольного параллелепипеда  
def calculate_parallelepiped_properties(a, b, c):  
    # Вычисление объема  
    volume = a * b * c  
    # Вычисление площади поверхности  
    surface_area = 2 * (a * b + b * c + a * c)  
    return volume, surface_area  
  
# Ввод длины ребер  
a = float(input("Введите длину ребра a: "))  
b = float(input("Введите длину ребра b: "))  
c = float(input("Введите длину ребра c: "))  
  
# Вызов функции и получение результатов  
volume, surface_area = calculate_parallelepiped_properties(a, b, c)  
  
# Вывод результатов  
print(f"Объем прямоугольного параллелепипеда: {volume}")  
print(f"Площадь поверхности прямоугольного параллелепипеда: {surface_area}")
```

Как работает программа:

1. Определена функция `calculate_parallelepiped_properties`, которая принимает длины трех ребер:
 - * (a)
 - * (b)
 - * (c)
2. Функция вычисляет объем ($V = abc$) и площадь поверхности ($S = 2(ab + bc + ac)$) параллелепипеда.
3. Пользователь вводит значения для (a), (b) и (c).
4. После этого программа вызывает функцию для получения объема и площади поверхности.
5. Результаты выводятся на экран.

Ниже представлена программа, которая на языке Питон вычисляет объем и площадь поверхности прямоугольного параллелепипеда, используя заданные длины его ребер

Задача 1

```
python Copy code  
  
# функция для вычисления объема и поверхности прямоугольного параллелепипеда  
def calculate_parallelepiped_properties(a, b, c):  
    # Вычисление объема  
    volume = a * b * c  
    # Вычисление площади поверхности  
    surface_area = 2 * (a * b + b * c + a * c)  
    return volume, surface_area  
  
# Ввод длины ребер  
a = float(input("Введите длину ребра a: "))  
b = float(input("Введите длину ребра b: "))  
c = float(input("Введите длину ребра c: "))  
  
# Вызов функции и получение результатов  
volume, surface_area = calculate_parallelepiped_properties(a, b, c)  
  
# Вывод результатов  
print(f"Объем прямоугольного параллелепипеда: {volume}")  
print(f"Площадь поверхности прямоугольного параллелепипеда: {surface_area}")
```

В результате объем прямоугольного параллелепипеда $V = 162$, а его площадь $S = 198$.

Задача 2

Вы можете использовать следующий код на языке Python для вычисления площади грани, площади полной поверхности и объема куба, если длина ребра куба равна 12:

```
python Copy code  
  
# Длина ребра куба  
a = 12  
  
# Площадь грани куба  
s1 = a ** 2  
  
# Площадь полной поверхности куба  
s2 = 6 * a ** 2  
  
# Объем куба  
v = a ** 3  
  
# Вывод результатов  
print(f"Площадь грани куба (s1): {s1}")  
print(f"Площадь полной поверхности куба (s2): {s2}")  
print(f"Объем куба (v): {v}")
```

Когда вы выполните этот код, он выведет значения для площади грани, площади полной поверхности и объема куба.

Задача 2

```
python Copy code

# Длина ребра куба
a = 12

# Площадь грани (S1)
S1 = a**2

# Площадь полной поверхности (S2)
S2 = 6 * a**2

# Объем куба (V)
V = a**3

# Вывод результатов
print(f"Длина ребра куба (a): {a}")
print(f"Площадь грани (S1): {S1}")
print(f"Площадь полной поверхности (S2): {S2}")
print(f"Объем куба (V): {V}")
```

```
css Copy code

Длина ребра куба (a): 12
Площадь грани (S1): 144
Площадь полной поверхности (S2): 864
Объем куба (V): 1728
```

Ниже представлена программа, которая на языке Питон вычисляет площадь равнобедренной трапеции, используя заданные длины ее сторон.

Задача 3

```
python Copy code

1 import math
2
3 def trapezoid_area(a, b, c):
4     h = math.sqrt(c**2 - ((b - a) / 2)**2)
5     area = (a + b) / 2 * h
6     return area
7
8 # Значения a, b и c
9 a = 3
10 b = 5
11 c = 7
12
13 # Вычисление площади
14 area = trapezoid_area(a, b, c)
15 print(f"Площадь равнобедренной трапеции: {area}")
16
```

Ниже представлена программа, которая на языке Питон вычисляет площадь равнобедренной трапеции, используя заданные длины ее сторон.

Задача 3

Теперь подставим значения $a = 3$, $b = 5$ и $c = 7$ в программу:

1. Вычислим высоту:

$$h = \sqrt{7^2 - \left(\frac{5-3}{2}\right)^2} = \sqrt{49 - 1} = \sqrt{48} = 4\sqrt{3}$$

2. После этого можем найти площадь:

$$S = \frac{(3+5)}{2} \cdot 4\sqrt{3} = 4 \cdot 4\sqrt{3} = 16\sqrt{3}$$

Таким образом, площадь равнобедренной трапеции при заданных параметрах составляет $16\sqrt{3} \approx 27.71$.

Таким образом, площадь равнобедренной трапеции при заданных параметрах составляет примерно 27,71.

Чтобы определить, могут ли три числа быть длинами сторон треугольника, нужно использовать неравенство треугольника. Оно гласит, что сумма длин любых двух сторон должна быть больше длины третьей стороны. Это можно записать как:

$$1) a+b > c$$

$$2) a+c > b$$

$$3) b+c > a$$

Вот программа на языке Питон, которая реализует этот подход:

Задача 4

```
python Copy code  
  
1 def can_form_triangle(a, b, c):  
2     return (a + b > c) and (a + c > b) and (b + c > a)  
3  
4 # Ввод сторон треугольника  
5 a = float(input("Введите длину первой стороны: "))  
6 b = float(input("Введите длину второй стороны: "))  
7 c = float(input("Введите длину третьей стороны: "))  
8  
9 # Проверка  
10 if can_form_triangle(a, b, c):  
11     print("Эти числа могут быть длинами сторон треугольника.")  
12 else:  
13     print("Эти числа не могут быть длинами сторон треугольника.")  
14
```

Вот программа на языке Питон, которая проверяет, могут ли три числа быть длинами сторон треугольника, а также производит расчеты для указанных наборов значений: (5,7,9) и (11,32,5).

Задача 4

```
python Copy code  
  
1 def can_form_triangle(a, b, c):  
2     return (a + b > c) and (a + c > b) and (b + c > a)  
3  
4 # Вариант 1  
5 a1, b1, c1 = 5, 7, 9  
6 result1 = can_form_triangle(a1, b1, c1)  
7 print(f"При a={a1}, b={b1}, c={c1}: {'Могут' if result1 else 'Не могут'} быть  
8  
9 # Вариант 2  
10 a2, b2, c2 = 11, 32, 5  
11 result2 = can_form_triangle(a2, b2, c2)  
12 print(f"При a={a2}, b={b2}, c={c2}: {'Могут' if result2 else 'Не могут'} быть  
13
```

Задача 4

При выполнении данной программы, будет выведен результат для обоих наборов значений:

1. Для $a = 5$, $b = 7$, $c = 9$, проведя проверки:

$5+7>9$ (true)

$5+9>7$ (true)

$7+9>5$ (true)

Все условия выполнены, значит числа могут быть длинами сторон треугольника.

2. Для $a=11, b=32, c=5$, проведя проверки:

$11+32>5$ (true)

$11+5>3$ (false)

$32+5>11$ (true)

Второе условие не выполнено, значит числа не могут быть длинами сторон треугольника.

В результате выполнения программы будет выведено:

При $a=5, b=7, c=9$: могут быть длинами сторон треугольника.

При $a=11, b=32, c=5$: не могут быть длинами сторон треугольника.

В результате выполнения программы будет выведено:

При a=5, b=7, c=9: Могут быть длинами сторон треугольника.

При a=11, b=32, c=5: Не могут быть длинами сторон треугольника.

ОТВЕТЫ

№ задачи	Содержание	Результат
1.	Даны длины ребер a, b, c прямоугольного параллелепипеда. Найти его объем $V = abc$ и площадь поверхности $S = 2(ab + bc + ac)$.	<p>Окно вывода</p> <pre>162 198</pre> <p>Объем прямоугольного параллелепипеда: $V = 162$ Площадь поверхности прямоугольного параллелепипеда: $S = 198$</p>
2.	Дана длина ребра куба. Найти площадь грани ($S1=a^2$), площадь полной поверхности ($S2=6a^2$) и объем куба ($V = a^3$).	<p>Окно вывода</p> <pre>12 Площадь грани куба = 144 Площадь полной поверхности куба = 864 Объем куба = 1728</pre> <p>Длина ребра куба (a): 12 Площадь грани (S1): 144 Площадь полной поверхности (S2): 864 Объем куба (V): 1728</p>
3.	Составьте программу нахождения площади равнобедренной трапеции через стороны a, b, c.	<p>Окно вывода</p> <pre>3 5 7 Формула S = 27.712812921102</pre> <p>Площадь равнобедренной трапеции при заданных параметрах составляет примерно 27.71</p>
4.	Составить программу, которая по трем введенным вами числам определит, могут ли эти числа быть длинами сторон треугольника.	<p>Окно вывода</p> <pre>Введите значение переменных a, b, c 5 7 9 Да</pre> <p>Окно вывода</p> <pre>Введите значение переменных a, b, c 11 32 5 Нет</pre> <p>При a=5, b=7, c=9: Могут быть длинами сторон треугольника. При a=11, b=32, c=5: Не могут быть длинами сторон треугольника.</p>

6.Операторы цикла

Операторы цикла используются для вычислений, которые повторяются многократно. В Паскале имеется три вида циклов: цикл с предусловием **while**, цикл с постусловием **repeat** и цикл с параметром **for**.

6.1.Структурные схемы

Циклы с предусловием (на рис.1, левый) и с постусловием (рис.1, правый)

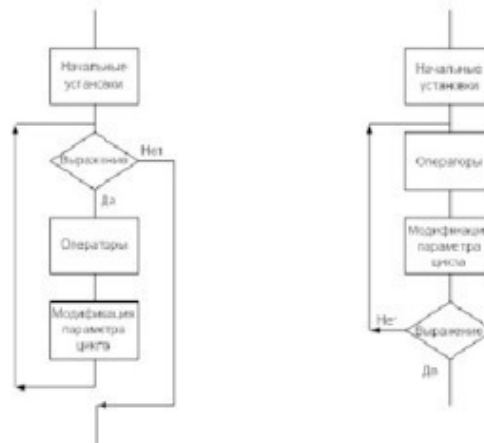


Рис.1.

Формат оператора с предусловием:

while выражение **do** оператор

Формат оператора с постусловием:

repeat тело цикла **until** выражение

6.2.Примеры

Программа печати таблицы значений функции **sin** для аргумента, изменяющегося в заданных пределах с заданным шагом.

```

program tabl_fun;
var
Xn, Xk: real; { начальное и конечное значения аргумента }
dX: real; { шаг изменения аргумента }
x, y: real; { текущие значения аргумента и функции }
begin
writeln («Введите Xn, Xk, dX»);
readln (Xn, Xk, dX);
writeln (« — — — — —»); { заголовок таблицы }
writeln (« | X | Y | »);

```

```
writeln (« — — — — —»);
x:= Xn; { первое значение аргумента }
while x <= Xk do { заголовок цикла }
begin
y:= sin (x); { вычисление значения функции }
writeln (« | », x:9:2, ' | », y:9:2, ' | »); { вывод строки таблицы }
x:= x+dX; { переход к следующему значению аргумента }
end;
writeln (« — — — — —»);
readln;
end.
```

Программа, вычисляющая квадратный корень вещественного аргумента X с заданной точностью eps по итерационной формуле:

$$y_n = (y_{n-1} + x/y_{n-1}) / 2,$$

где y_{n-1} – предыдущее приближение к корню (в начале вычислений выбирается произвольно), y_n – последующее приближение. Процесс вычислений прекращается, когда приближения станут отличаться друг от друга по абсолютной величине менее, чем на величину заданной точности.

```
program square_root;
var
X, eps, { аргумент и точность }
Yp, Y: real; { предыдущее и последующее приближение }
begin
repeat
writeln («Введите X, eps»);
readln (X, eps);
until (X> 0) and (eps> 0);
Y:= 1;
repeat
Yp:= Y;
Y:= (Yp + X / Yp) / 2;
until abs (Y – Yp) <eps;
writeln («Корень из», X:6:3, «с точностью», eps:7:5, «равен», Y:9:5);
readln;
```

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.