

Легковесная автоматическая проверка заданий LightPT – взгляд преподавателя

Бордюгова Т.Н.

*Южный федеральный университет,
Институт математики, механики и компьютерных наук*

Доклад на пятой Всероссийской научно-методической конференции
«Использование системы программирования PascalABC.NET
в обучении программированию»
(Ростов-на-Дону, 28-29 марта 2025 г.)

Автоматизированная проверка полезна для начинающих программистов по следующим ключевым причинам:

1. Обратная связь в режиме реального времени
2. Отсутствие зависимости от преподавателя
3. Умение решать проблемы
4. Повышение уровня внимания к деталям
5. Поддержание мотивации
6. Постепенное усложнение задач
7. Развитие хороших практик
8. Подготовка к реальной практике

Автоматизированная проверка создает благоприятные условия для быстрого освоения основ программирования, развития необходимых навыков и формирования хорошего отношения к обучению.

Несколько шагов, как можно эффективно внедрить автоматизированную проверку программ в учебный процесс:

1. Выбор платформы

Хотите ли вы проверять небольшие программы, код, написанный на уроке, или проводить полноценные соревнования?

2. Определение целей проверки

- Исправность работы программы (правильные входные данные должны давать правильные выходные).
- Эффективность выполнения (оптимизация по времени и памяти).
- Правильность структуры кода (например, использование правильных переменных, функций и методов).

Несколько шагов, как можно эффективно внедрить автоматизированную проверку программ в учебный процесс:

3. Подготовка тестовых случаев

- Простые входные данные (для проверки базовой функциональности).
- Угловые случаи (некорректные входные данные, граничные значения, пустые массивы/строки и т.п.).
- Сильно нагружающие тесты (для проверки производительности).

4. Интеграция заданий в уроки

- Домашние задания.
- Лабораторные работы.
- Контрольные работы.

Несколько шагов, как можно эффективно внедрить автоматизированную проверку программ в учебный процесс:

6. Анализ результатов

Учителя могут отслеживать прогресс учеников, выявляя общие проблемы или слабые места в понимании темы.

7. Соревнования и мотивация

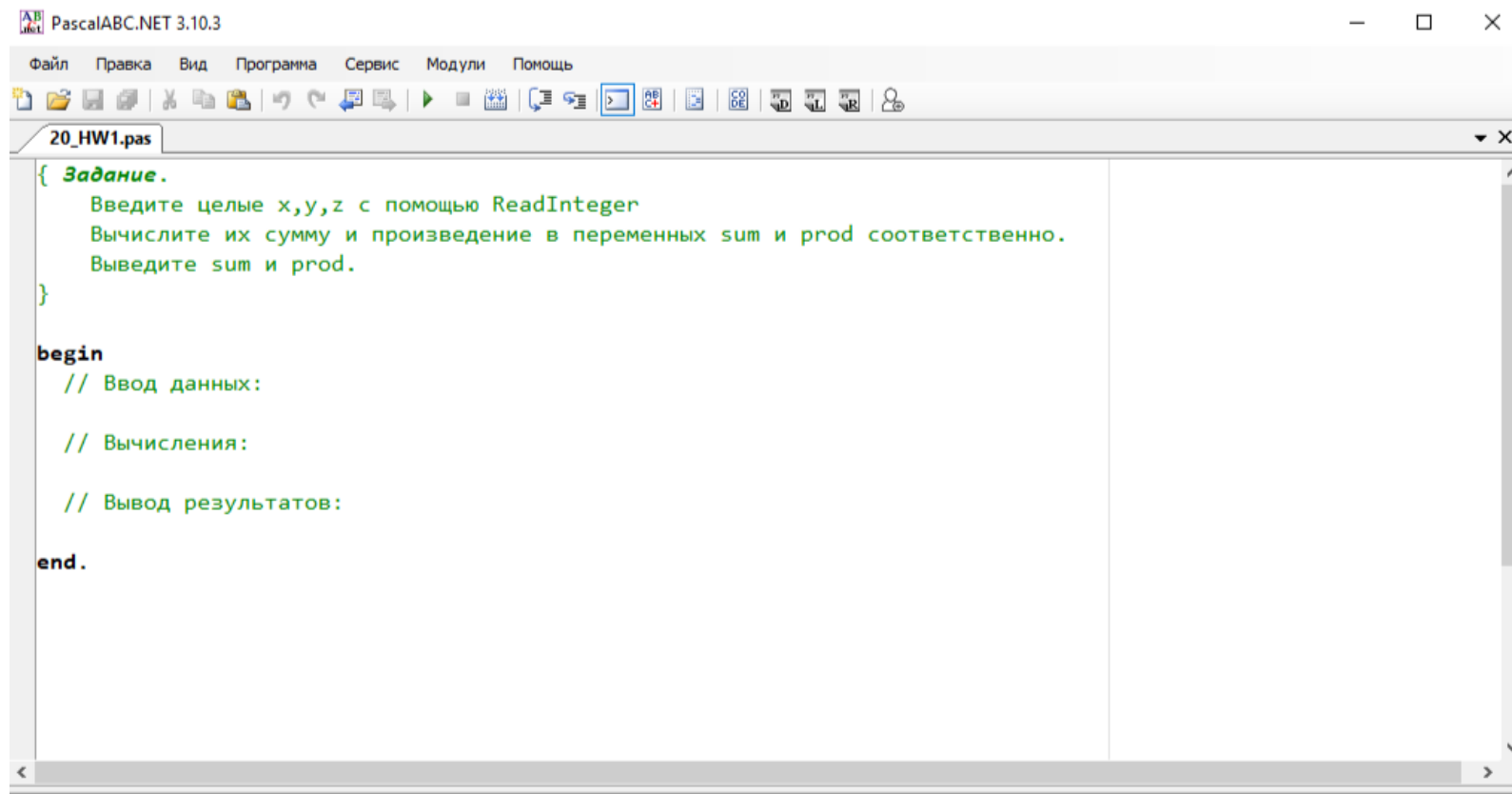
Чтобы поддерживать интерес школьников, можно устраивать мини-соревнования по решению задач. .

Особенности нового модуля проверки LightPT

- Задания необходимо составлять к каждому занятию. Тесты составляются вместе с заданиями, то есть **оперативно**
- **Низкая сложность** составления тестов (одна-две строки на простые задачи)
- В отличие от олимпиадных систем осуществляется **типизированная проверка данных**
- Гибкое управление вариантами входных и выходных данных: данные могут **вводиться** или **генерироваться** случайно (в олимпиадных системах невозможно)
- Возможность вводить и выводить данные несколькими способами, в разном порядке, выводить **промежуточные** данные
- Подключаемые модули проверки скрыты – **невидимы** ученику
- Возможность **интеллектуальных подсказок** при частичном решении
- Некоторые выводимые данные могут **игнорироваться при анализе решения**
- Лёгкое сохранение результатов запуска в БД и визуализация результатов группы учеников
- **Самое главное** – ученик может начинать с **пустой программы**. У него складывается полное ощущение, что он решает сам, а кто-то невидимый ему подсказывает и проверяет

Простейший пример

- 8 класс. Ученику предлагается задача – заданы три целых числа, найти их сумму и произведение. Предлагается следующий пустой шаблон программы:



```
PascalABC.NET 3.10.3
Файл  Правка  Вид  Программа  Сервис  Модули  Помощь
[Icons]
20_HW1.pas
{ Задание.
  Введите целые x,y,z с помощью ReadInteger
  Вычислите их сумму и произведение в переменных sum и prod соответственно.
  Выведите sum и prod.
}

begin
  // Ввод данных:

  // Вычисления:

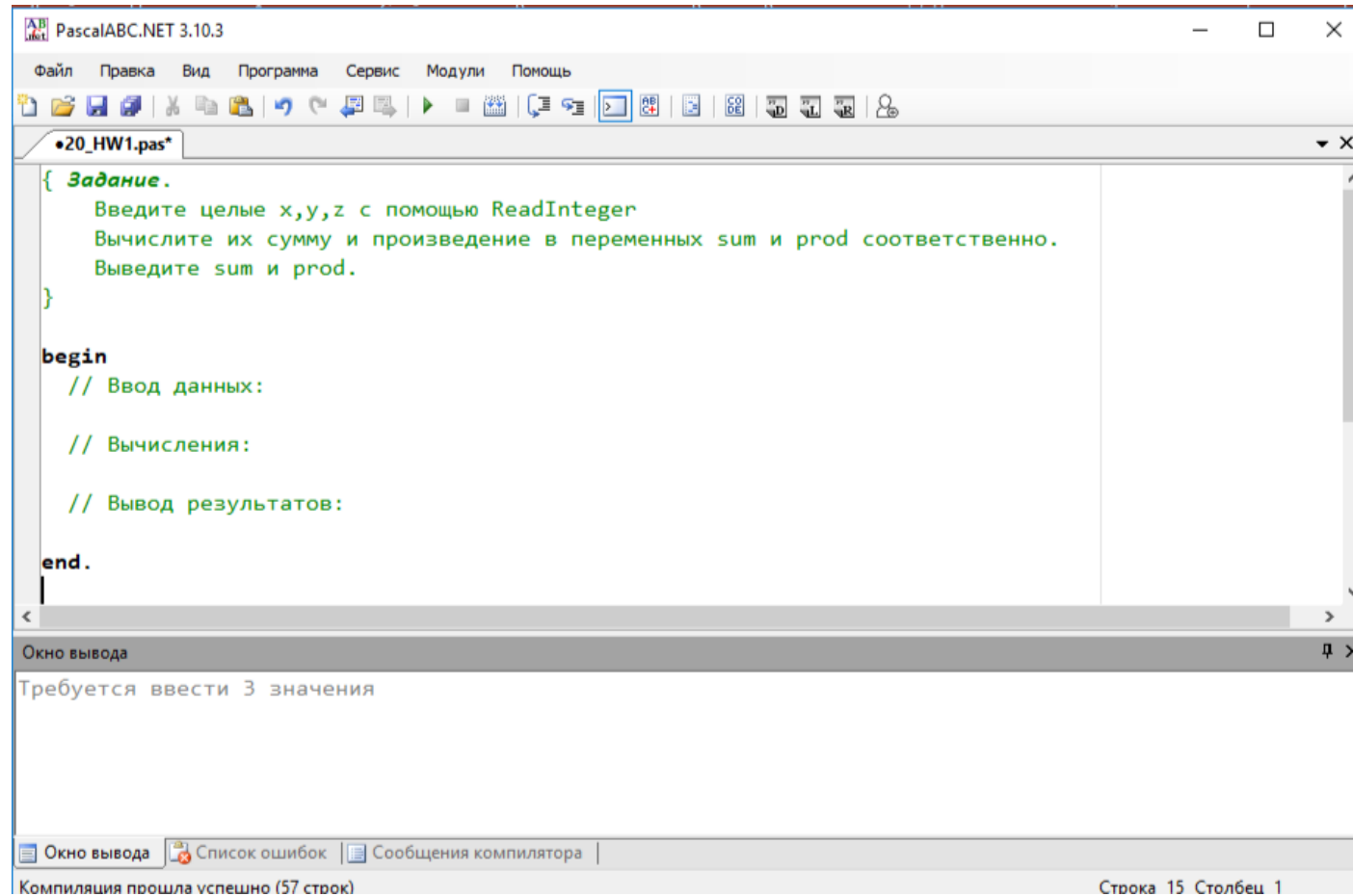
  // Вывод результатов:

end.
```

- Ученик может решать задачу **в свободной форме**

Запуск

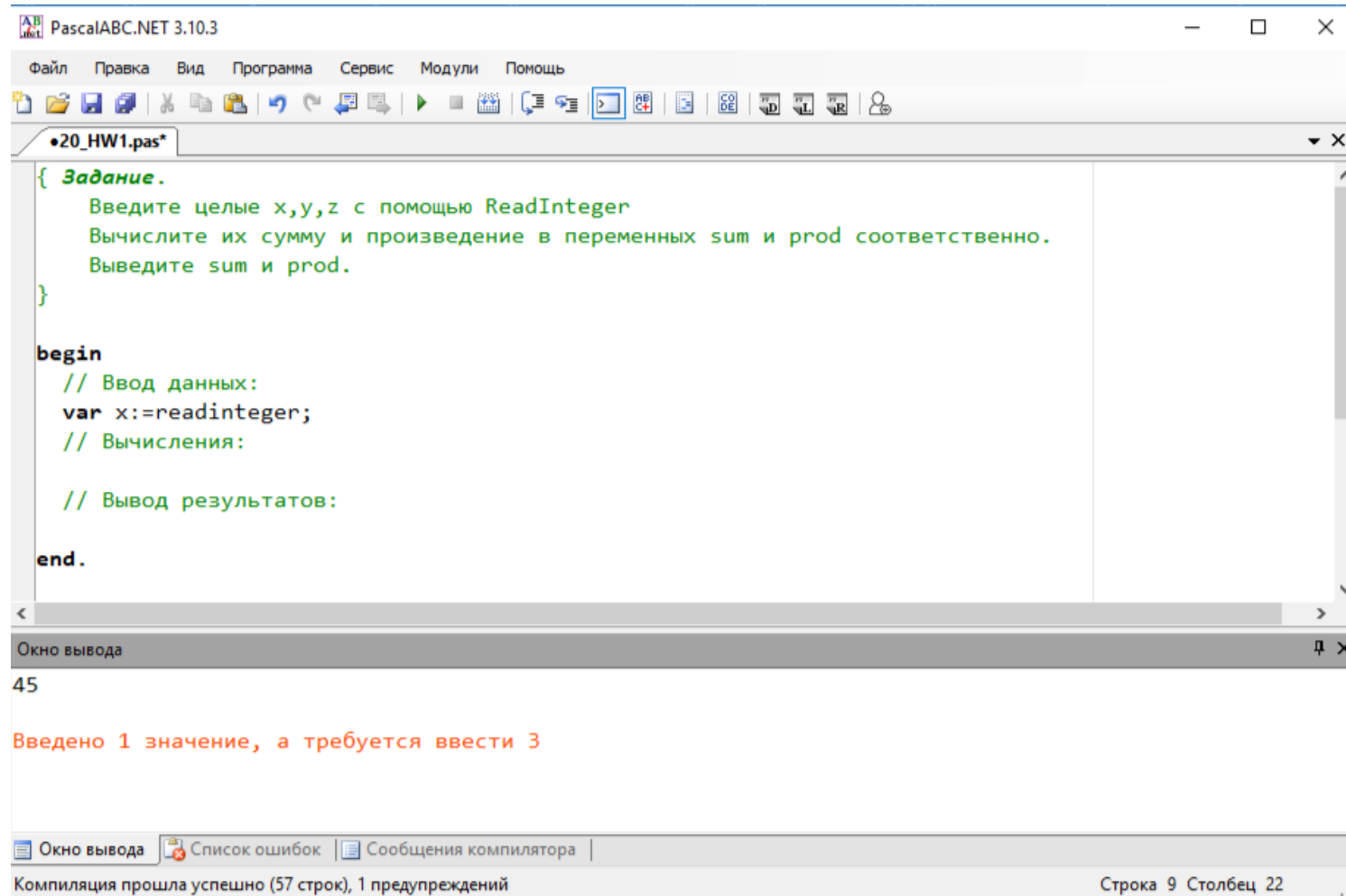
- Ученик запускает программу:



- Подсказка внизу сообщает, что нужно сделать

Ввод

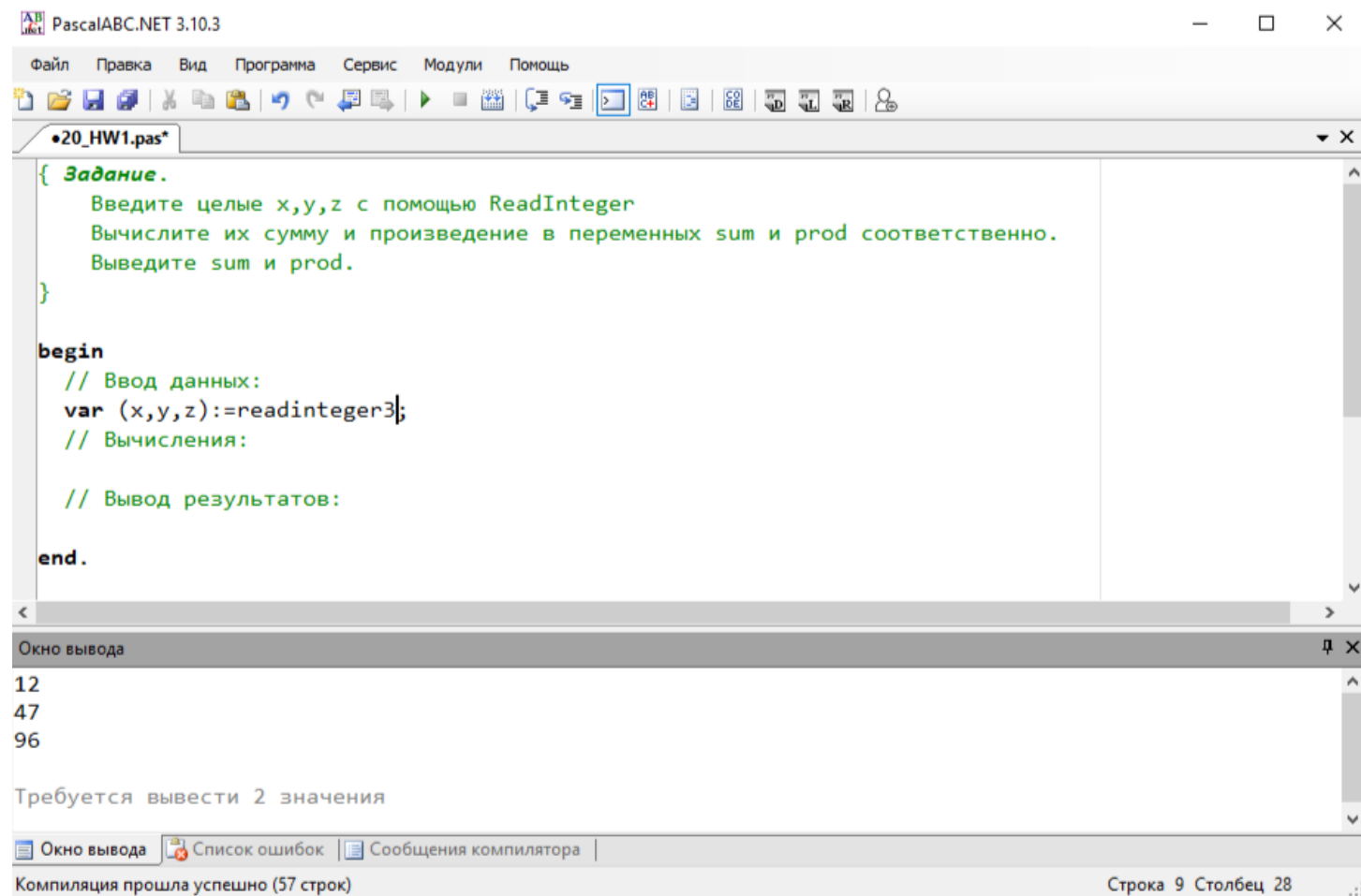
- Ученик вводит одно значение



- Подсказка внизу сообщает, что нужно сделать

Ввод 2

- Ученик вводит три значения



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main window displays a Pascal program named '20_HW1.pas'. The program includes a task description in Russian, followed by a 'begin' block with comments and code for reading three integers, calculating their sum and product, and printing the results. The 'end.' statement concludes the program. Below the code editor, the 'Окно вывода' (Output Window) is visible, showing the output of the program: '12', '47', and '96'. A message 'Требуется вывести 2 значения' (Need to output 2 values) is also present. The status bar at the bottom indicates 'Компиляция прошла успешно (57 строк)' (Compilation successful (57 lines)) and 'Строка 9 Столбец 28' (Line 9 Column 28).

```
{ Задание.  
  Введите целые x,y,z с помощью ReadInteger  
  Вычислите их сумму и произведение в переменных sum и prod соответственно.  
  Выведите sum и prod.  
}  
  
begin  
  // Ввод данных:  
  var (x,y,z):=readinteger3;  
  // Вычисления:  
  
  // Вывод результатов:  
  
end.
```

Окно вывода

12
47
96

Требуется вывести 2 значения

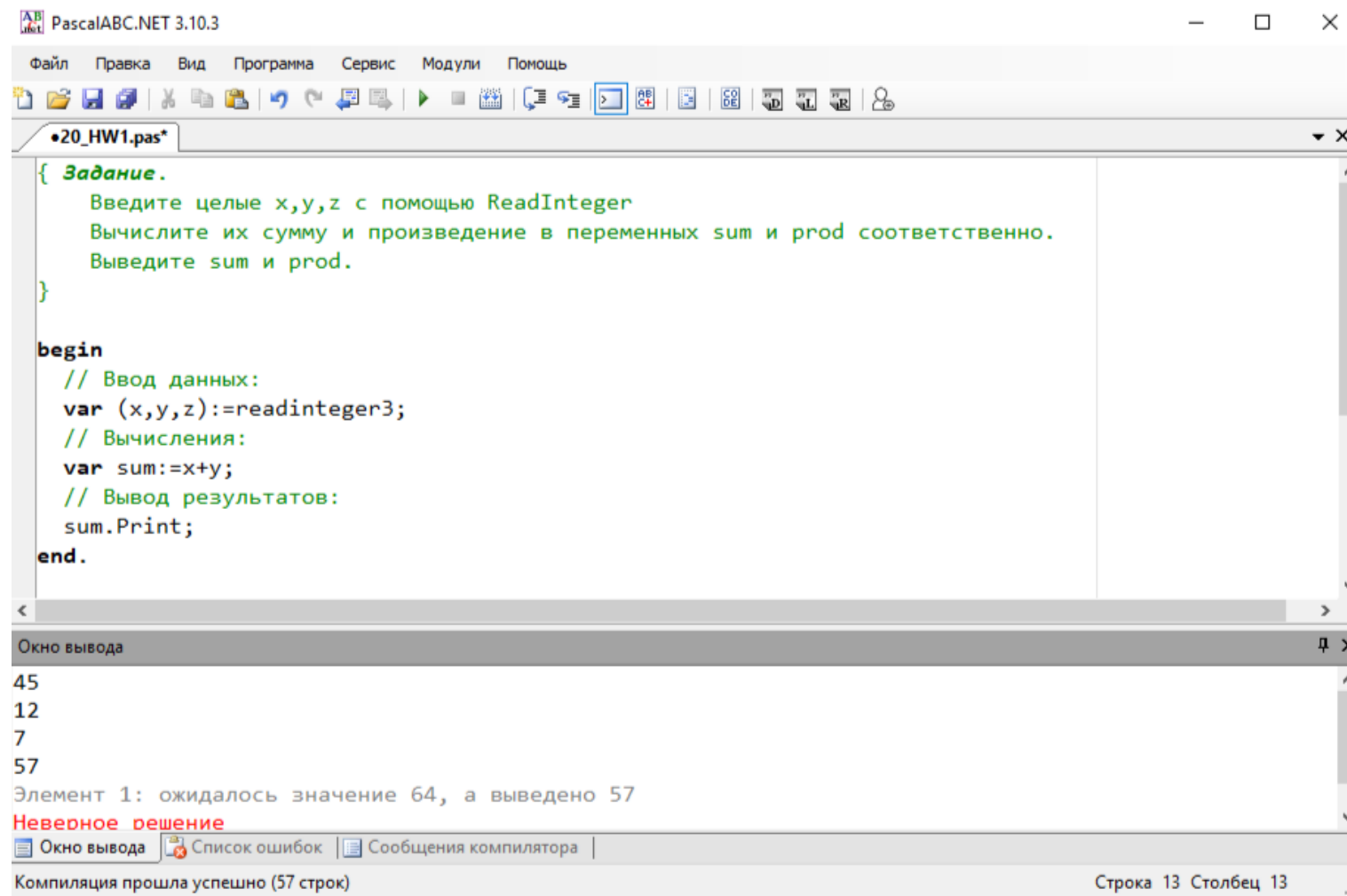
Окно вывода | Список ошибок | Сообщения компилятора

Компиляция прошла успешно (57 строк) Строка 9 Столбец 28

- Теперь нужно решить задачу и вывести 2 значения

Частичное решение (неверное)

- Ученик пытается решить задачу



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main window displays a Pascal program named 20_HW1.pas. The program is a simple addition task. The output window shows the results of the program's execution.

```
{ Задание.  
  Введите целые x,y,z с помощью ReadInteger  
  Вычислите их сумму и произведение в переменных sum и prod соответственно.  
  Выведите sum и prod.  
}  
  
begin  
  // Ввод данных:  
  var (x,y,z):=readinteger3;  
  // Вычисления:  
  var sum:=x+y;  
  // Вывод результатов:  
  sum.Print;  
end.
```

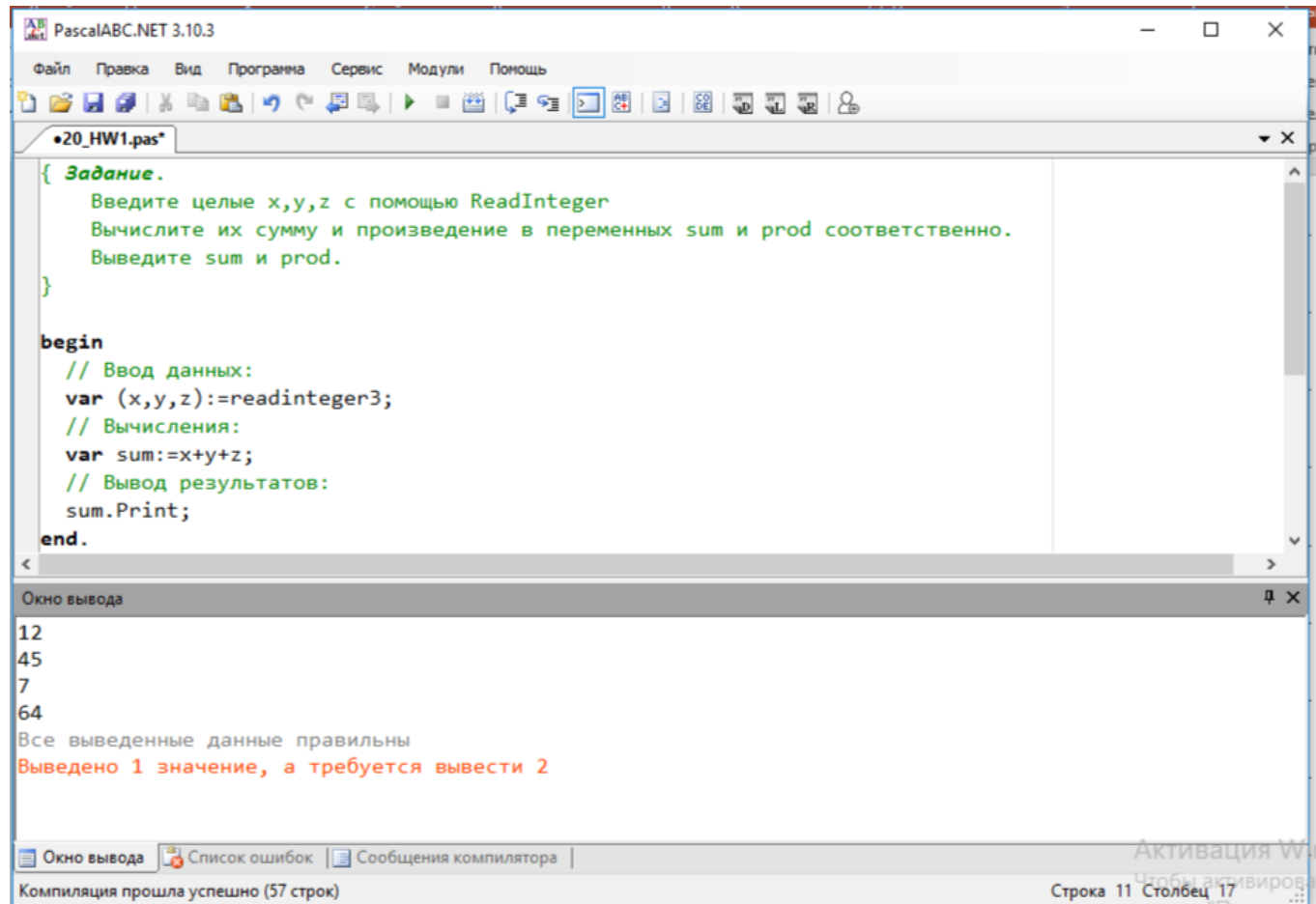
Окно вывода

```
45  
12  
7  
57  
Элемент 1: ожидалось значение 64, а выведено 57  
Неверное решение  
Окно вывода | Список ошибок | Сообщения компилятора  
Компиляция прошла успешно (57 строк) | Строка 13 Столбец 13
```

- Теперь нужно подумать, что не?

Частичное решение (верное)

- Ученик исправляет решение



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main window displays a Pascal program named '20_HW1.pas'. The program includes a task description in Russian and a solution. The task asks for the sum and product of three integers x, y, and z. The solution uses the `ReadInteger` function to read three integers, calculates their sum, and prints it. The output window shows the results of the program execution: 12, 45, 7, and 64. A message at the bottom of the output window states: 'Все выведенные данные правильны' (All output data is correct) and 'Выведено 1 значение, а требуется вывести 2' (1 value was output, but 2 are required). The status bar at the bottom indicates 'Компиляция прошла успешно (57 строк)' (Compilation successful (57 lines)) and 'Строка 11 Столбец 17' (Line 11, Column 17).

```
{ Задание.  
  Введите целые x,y,z с помощью ReadInteger  
  Вычислите их сумму и произведение в переменных sum и prod соответственно.  
  Выведите sum и prod.  
}  
  
begin  
  // Ввод данных:  
  var (x,y,z):=readinteger3;  
  // Вычисления:  
  var sum:=x+y+z;  
  // Вывод результатов:  
  sum.Print;  
end.
```

Окно вывода

12
45
7
64
Все выведенные данные правильны
Выведено 1 значение, а требуется вывести 2

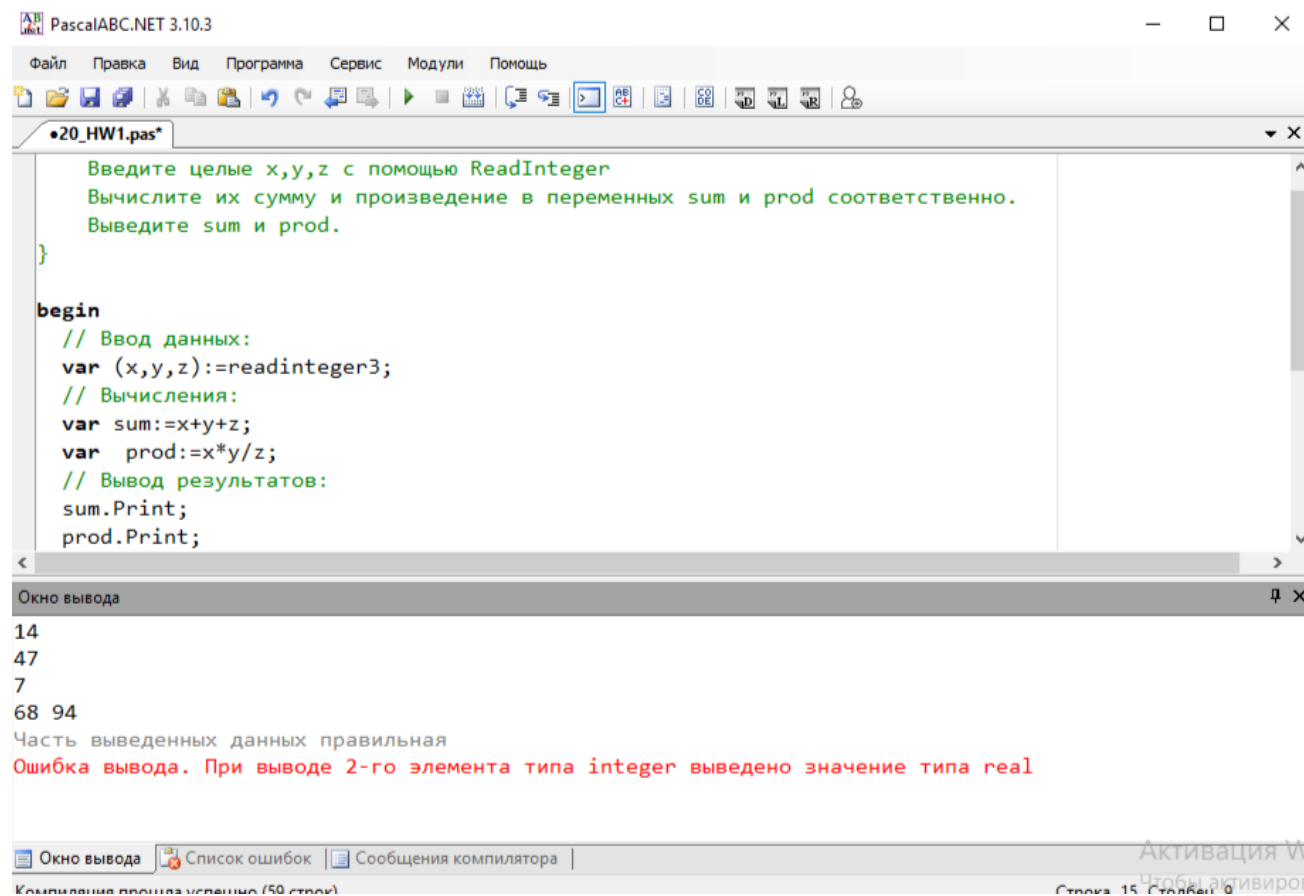
Окно вывода | Список ошибок | Сообщения компилятора

Компиляция прошла успешно (57 строк) Строка 11 Столбец 17

- Так, первое значение найдено верно – дело за вторым.

Частичное решение (верное)

- Попробуем обмануть! Выведем что-нибудь



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main editor window displays a Pascal program named `20_HW1.pas`. The program's logic is as follows: it prompts the user to enter three integers `x, y, z` using `ReadInteger`, calculates their sum (`sum = x + y + z`) and product (`prod = x * y / z`), and then prints both results. The output window at the bottom shows the execution results. It displays the values `14`, `47`, `7`, and `68 94` on separate lines. Below these, a message states: "Часть выведенных данных правильная" (Part of the output data is correct). A red error message follows: "Ошибка вывода. При выводе 2-го элемента типа integer выведено значение типа real" (Output error. When outputting the 2nd element of type integer, a value of type real was output). The status bar at the bottom indicates "Компиляция прошла успешно (50 строк)" (Compilation successful (50 lines)).

```
Введите целые x,y,z с помощью ReadInteger
Вычислите их сумму и произведение в переменных sum и prod соответственно.
Выведите sum и prod.
}

begin
  // Ввод данных:
  var (x,y,z):=readinteger3;
  // Вычисления:
  var sum:=x+y+z;
  var prod:=x*y/z;
  // Вывод результатов:
  sum.Print;
  prod.Print;
```

Окно вывода

14
47
7
68 94
Часть выведенных данных правильная
Ошибка вывода. При выводе 2-го элемента типа integer выведено значение типа real

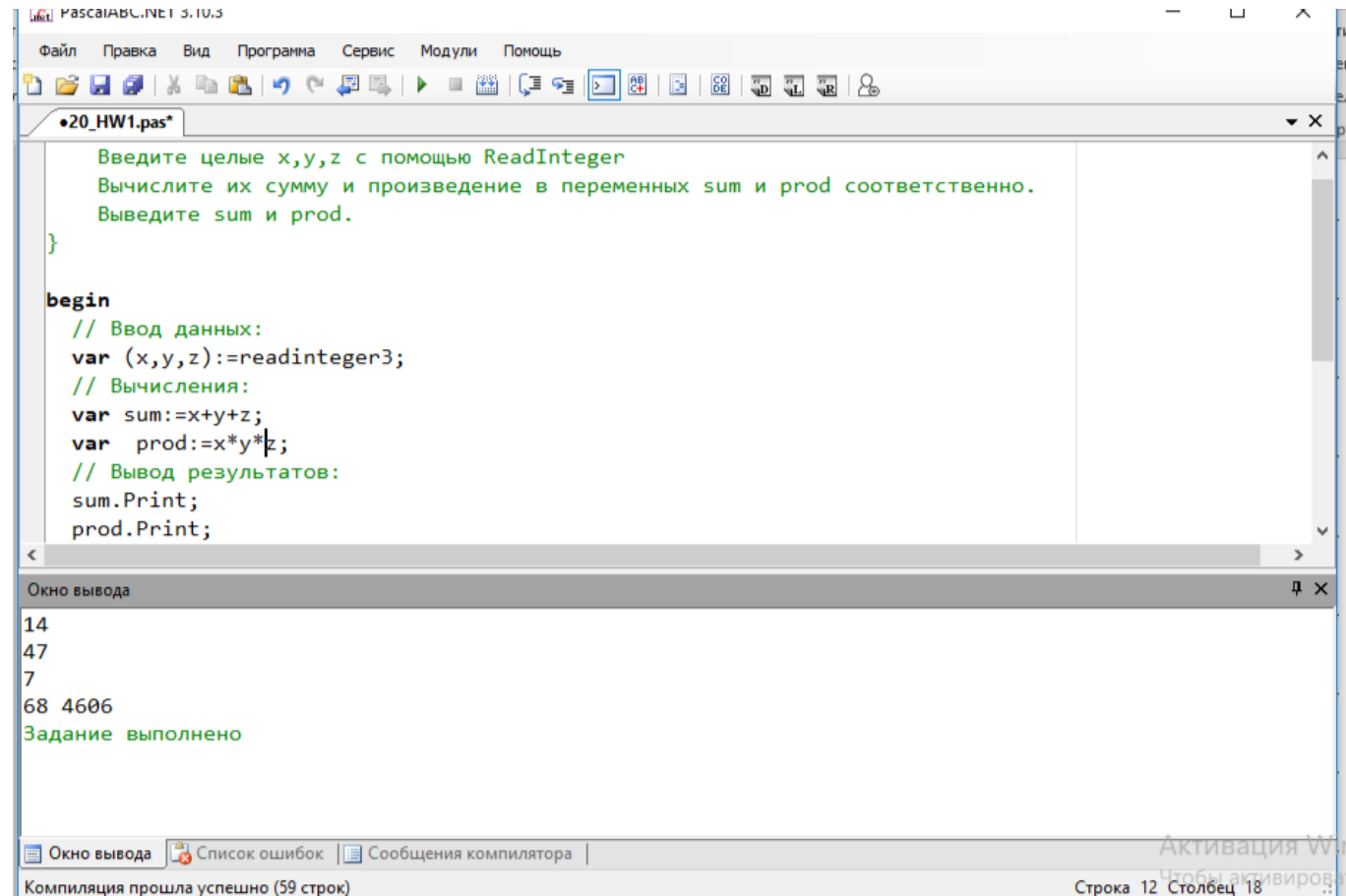
Окно вывода | Список ошибок | Сообщения компилятора |

Компиляция прошла успешно (50 строк) | Строчка 15 | Столбец 0

- Неверный тип при выводе!!! Система проверяет типы!

Полное решение

- Ученик решает задачу



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main editor window displays a Pascal program named '20_HW1.pas'. The program prompts the user to enter three integers x, y, and z, calculates their sum and product, and prints the results. The output window at the bottom shows the execution results: 14, 47, 7, 68, 4606, and a green message 'Задание выполнено' (Task completed). The status bar at the bottom indicates 'Компиляция прошла успешно (59 строк)' (Compilation successful (59 lines)).

```
Введите целые x,y,z с помощью ReadInteger
Вычислите их сумму и произведение в переменных sum и prod соответственно.
Выведите sum и prod.
}

begin
    // Ввод данных:
    var (x,y,z):=readinteger3;
    // Вычисления:
    var sum:=x+y+z;
    var prod:=x*y*z;
    // Вывод результатов:
    sum.Print;
    prod.Print;
```

Окно вывода

```
14
47
7
68 4606
Задание выполнено
```

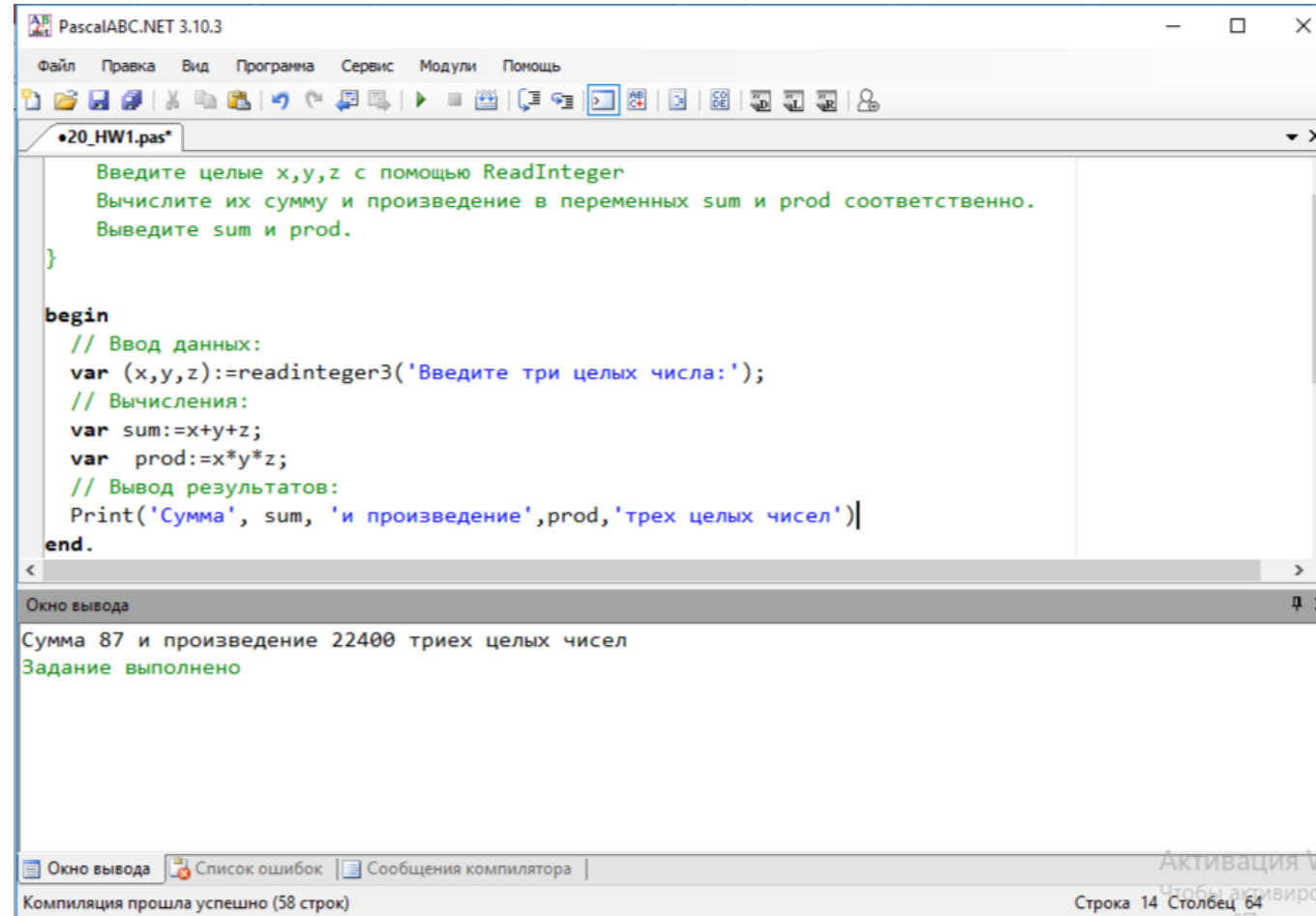
Окно вывода | Список ошибок | Сообщения компилятора

Компиляция прошла успешно (59 строк) | Строка 12 Столбец 18

- Наконец – заветное **Задание выполнено!**

Вариант 1

- Ввод может содержать приглашение к вводу, а вывод – пояснение. Они не учитываются при проверке!



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main window displays a Pascal program named '20_HW1.pas'. The program includes comments in Russian and Pascal code that reads three integers, calculates their sum and product, and prints the results. The output window at the bottom shows the execution results: 'Сумма 87 и произведение 22400 трех целых чисел' and 'Задание выполнено'. The status bar at the bottom indicates 'Компиляция прошла успешно (58 строк)' and 'Строка 14 Столбец 64'.

```
Введите целые x,y,z с помощью ReadInteger
Вычислите их сумму и произведение в переменных sum и prod соответственно.
Выведите sum и prod.
}

begin
  // Ввод данных:
  var (x,y,z):=readinteger3('Введите три целых числа:');
  // Вычисления:
  var sum:=x+y+z;
  var prod:=x*y*z;
  // Вывод результатов:
  Print('Сумма', sum, 'и произведение',prod,'трех целых чисел')
end.
```

Окно вывода

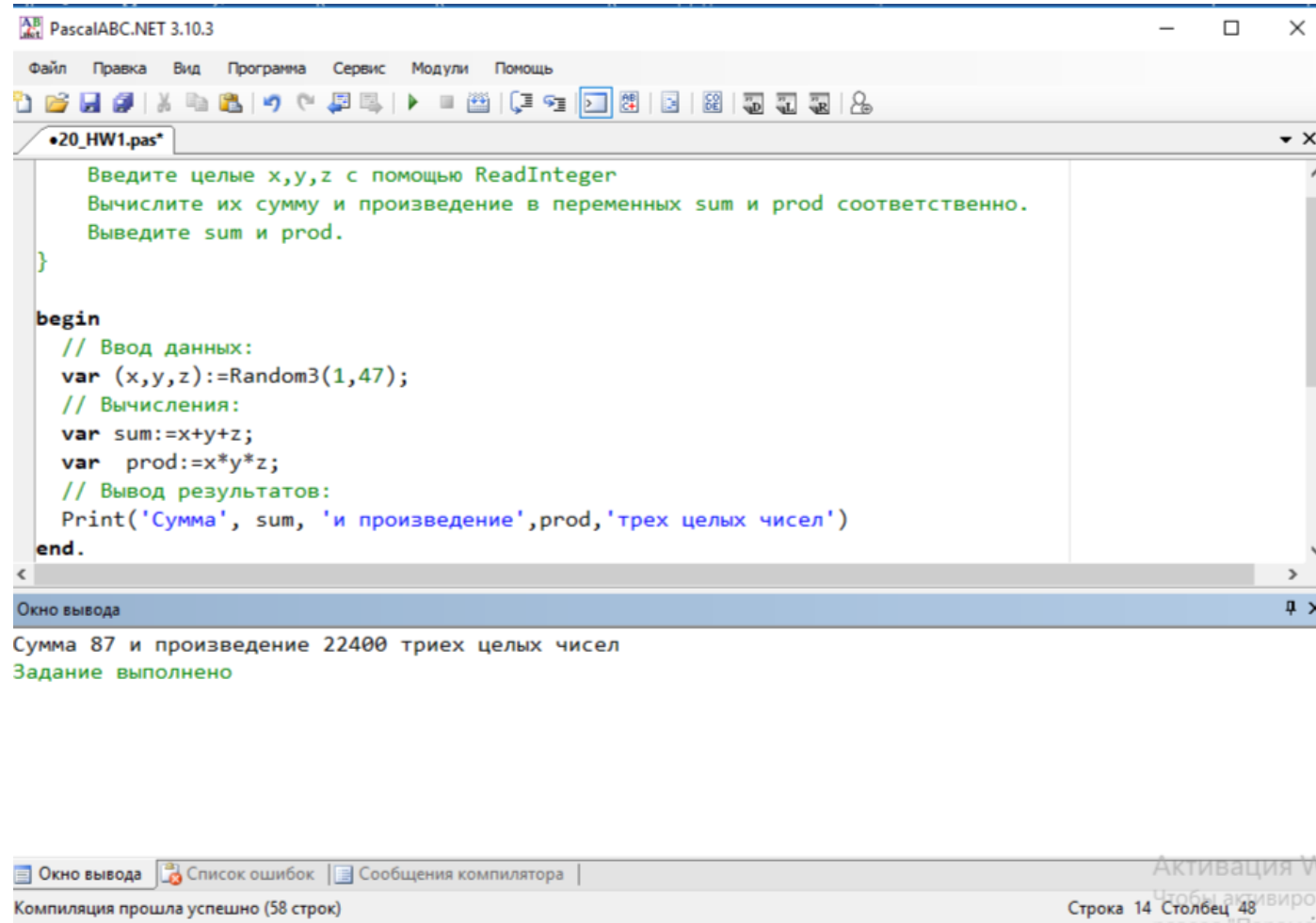
Сумма 87 и произведение 22400 трех целых чисел
Задание выполнено

Окно вывода | Список ошибок | Сообщения компилятора | Активация V...
Компиляция прошла успешно (58 строк) | Строка 14 Столбец 64

- Таким образом обеспечивается свобода ученика при решении задачи

Вариант 2

- Вместо данных, вводимых с клавиатуры, мы можем использовать **заполнение случайными**:



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main window displays a Pascal program named '20_HW1.pas'. The program uses the `Random3` function to generate three random integers between 1 and 47, calculates their sum and product, and prints the results. The output window shows the results: 'Сумма 87 и произведение 22400 трех целых чисел' and 'Задание выполнено'. The status bar at the bottom indicates 'Компиляция прошла успешно (58 строк)' and 'Строка 14 Столбец 48'.

```
Введите целые x,y,z с помощью ReadInteger
Вычислите их сумму и произведение в переменных sum и prod соответственно.
Выведите sum и prod.
}

begin
    // Ввод данных:
    var (x,y,z):=Random3(1,47);
    // Вычисления:
    var sum:=x+y+z;
    var prod:=x*y*z;
    // Вывод результатов:
    Print('Сумма', sum, 'и произведение',prod,'трех целых чисел')
end.
```

Окно вывода

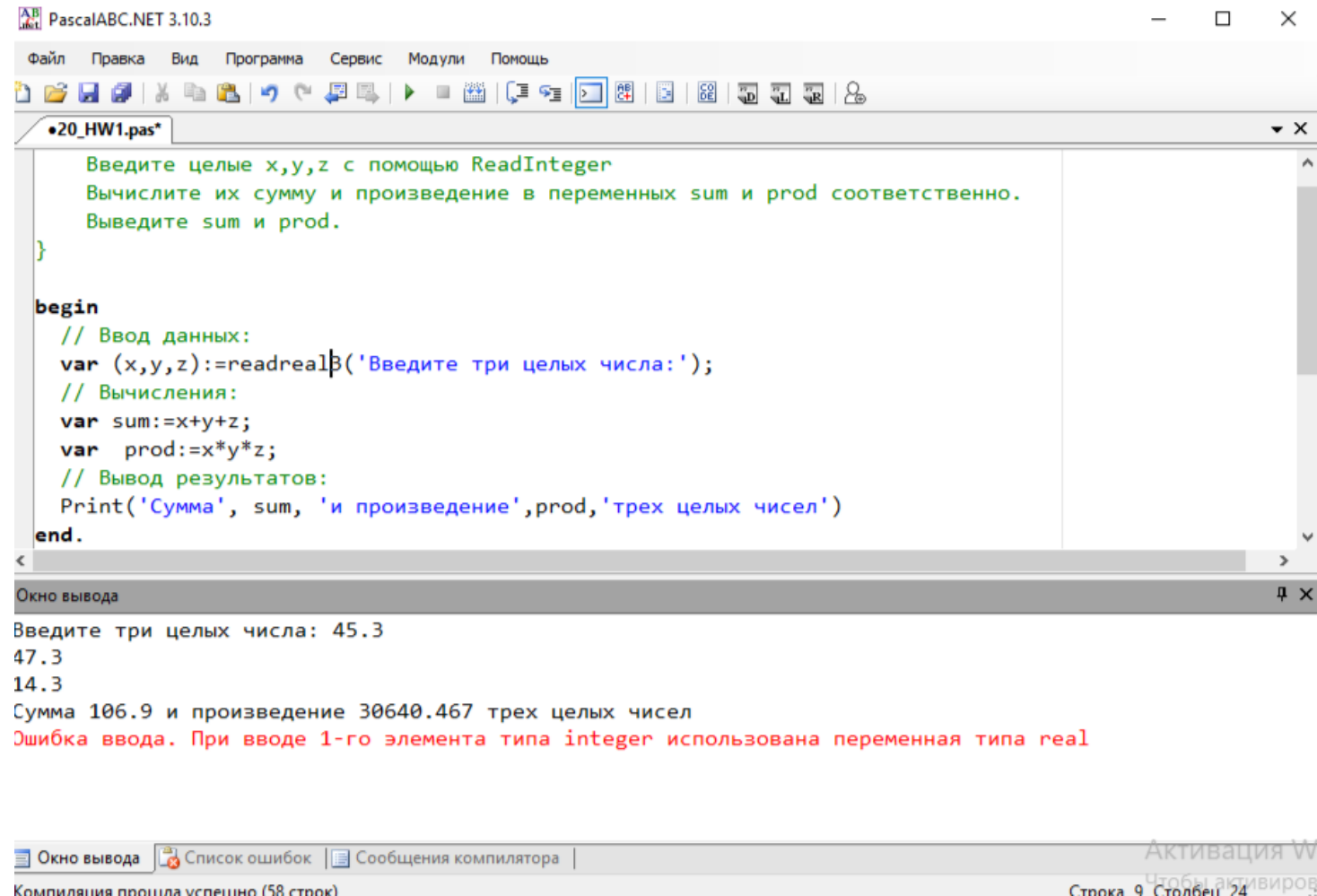
Сумма 87 и произведение 22400 трех целых чисел
Задание выполнено

Окно вывода | Список ошибок | Сообщения компилятора | Активация W...
Компиляция прошла успешно (58 строк) | Строка 14 Столбец 48

- Таким образом обеспечивается свобода ученика при решении задачи
«Модуль невидимой автоматической проверки заданий в среде PascalABC.NET»

Вариант 3. Другие типы данных

- Ученик может решить работать с вещественными числами



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main window displays a Pascal program named '20_HW1.pas'. The program prompts the user to enter three integers (x, y, z) using `ReadInteger`, calculates their sum and product, and prints the results. The output window shows the execution results, including a runtime error: 'Ошибка ввода. При вводе 1-го элемента типа integer использована переменная типа real' (Input error. When entering the 1st element of type integer, a variable of type real was used).

```
Введите целые x,y,z с помощью ReadInteger
Вычислите их сумму и произведение в переменных sum и prod соответственно.
Выведите sum и prod.
}

begin
  // Ввод данных:
  var (x,y,z):=readreal('Введите три целых числа:');
  // Вычисления:
  var sum:=x+y+z;
  var prod:=x*y*z;
  // Вывод результатов:
  Print('Сумма', sum, 'и произведение', prod, 'трех целых чисел')
end.
```

Окно вывода

Введите три целых числа: 45.3
47.3
14.3
Сумма 106.9 и произведение 30640.467 трех целых чисел
Ошибка ввода. При вводе 1-го элемента типа integer использована переменная типа real

Окно вывода | Список ошибок | Сообщения компилятора | Активация Windows

Компиляция прошла успешно (58 строк) | Строка 9 Столбец 24

Как создать такое проверяемое задание?

- Запустить скрипт ZCreateTasksNew.pas в папке с заданиями.
- Создастся файл Tasks0.pas, который надо переименовать в **Tasks.pas**
- В Tasks.pas в функцию **CheckTaskT** в ветку оператора case, соответствующую задаче, добавить код проверки:

```
• case name of
  'HW1': begin
    CheckData(InitialInput := |cInt,cInt,cInt|);
    GenerateTestData := tnum -> begin
      var (a,b,c) := Random3(1,100);
      InputList.AddTestData(|a,b,c|);
    end;
    CheckOutput(Int(0)+Int(1)+Int(2), Int(0)*Int(1)*Int(2));
  end;
  ...
```

Проверить количество введенных данных

Сравнить вывод с предполагаемым

Разработка задания на некоторые простые проверки

Учитель разрабатывает задание по теме «Программирование линейных алгоритмов» 8 класс

Задание.

Введите вещественное x

Вычислите $x*x$, $x*x*x$ и $x*x*x*x$ в переменных $x2$, $x3$ и $x4$ соответственно

Выведите $x2$, $x3$ и $x4$

$x = 4.1234$

$x2 = 17.00242756$ $x3 = 70.107809800904$ $x4 = 289.082542933048$

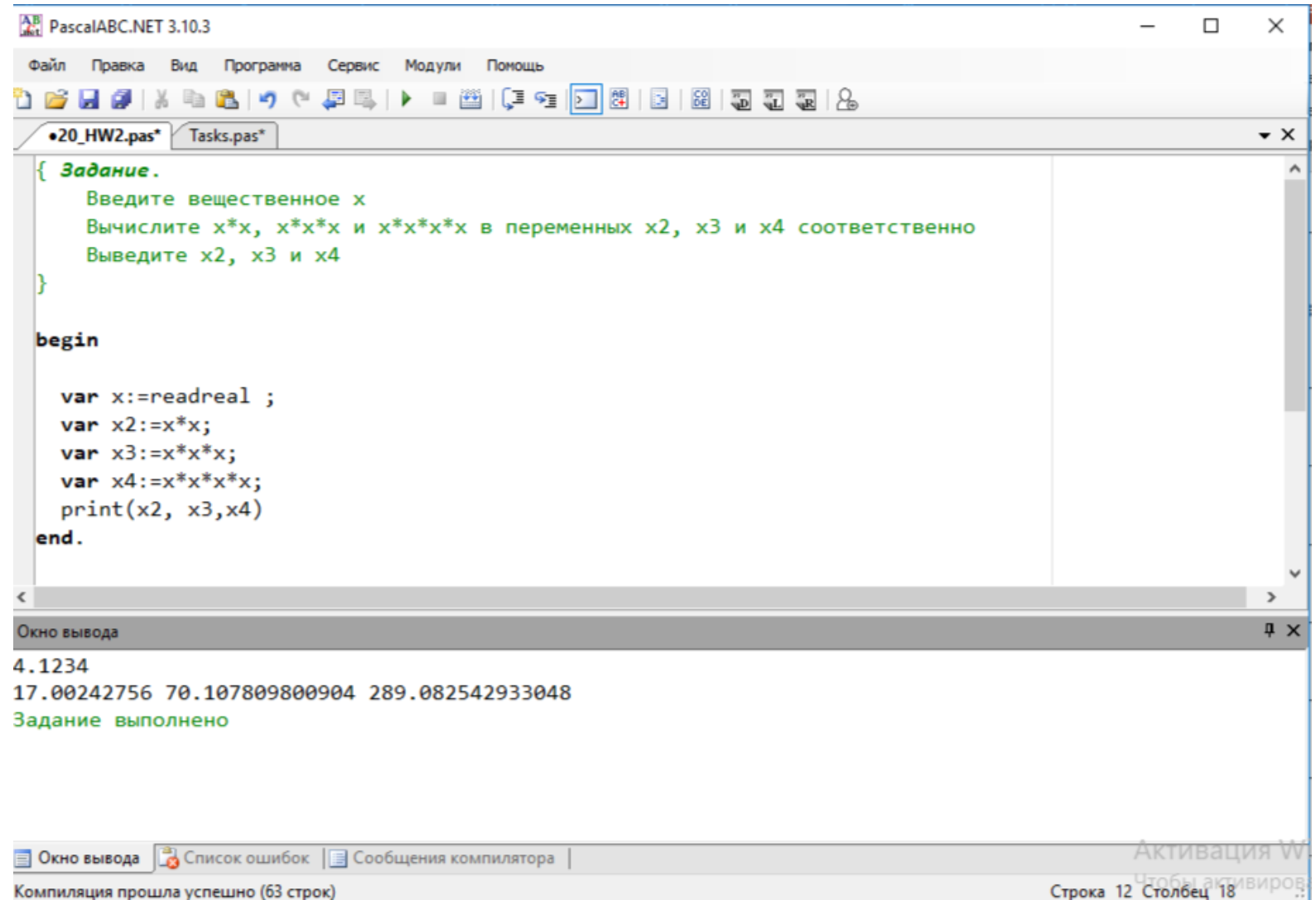
Разработка задания на некоторые простые проверки

Учитель создаёт проверочную часть в модуле **Tasks.pas**:

```
case name of
  'HW2': begin
    FilterOnlyNumbers;
    CheckData(Input := |cRe|);
    var x := Re(0);
    TestCount := 10;
    GenerateTestData := tnum -> begin // это когда есть особый случай
    var a: real;
    if tnum = 5 then
      a := RandomReal(1,100,digits := 0)
    else a := RandomReal(1,10,digits := 0);
    InputList.AddTestData(|a|);
    end;
    CheckOutput(x*x,x*x*x,x*x*x*x);
  end;
...
end;
```

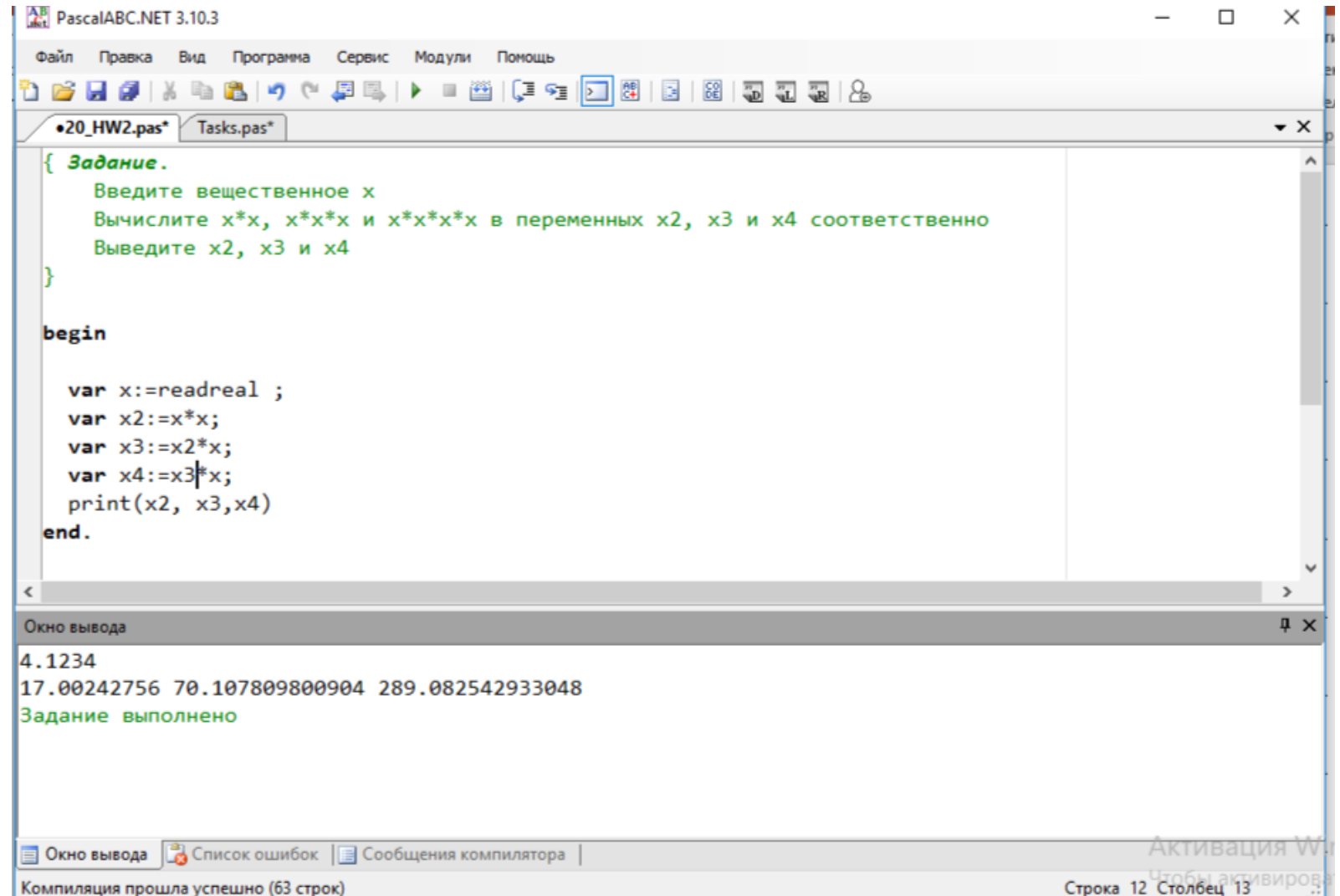
Разработка задания на некоторые простые проверки

Учитель запускает
решенное задание



Разработка задания на некоторые простые проверки

Учитель запускает
решенное задание



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main window displays a Pascal program named `20_HW2.pas`. The program code is as follows:

```
{ Задание.  
  Введите вещественное x  
  Вычислите x*x, x*x*x и x*x*x*x в переменных x2, x3 и x4 соответственно  
  Выведите x2, x3 и x4  
}  
  
begin  
  var x:=readreal ;  
  var x2:=x*x;  
  var x3:=x2*x;  
  var x4:=x3*x;  
  print(x2, x3,x4)  
end.
```

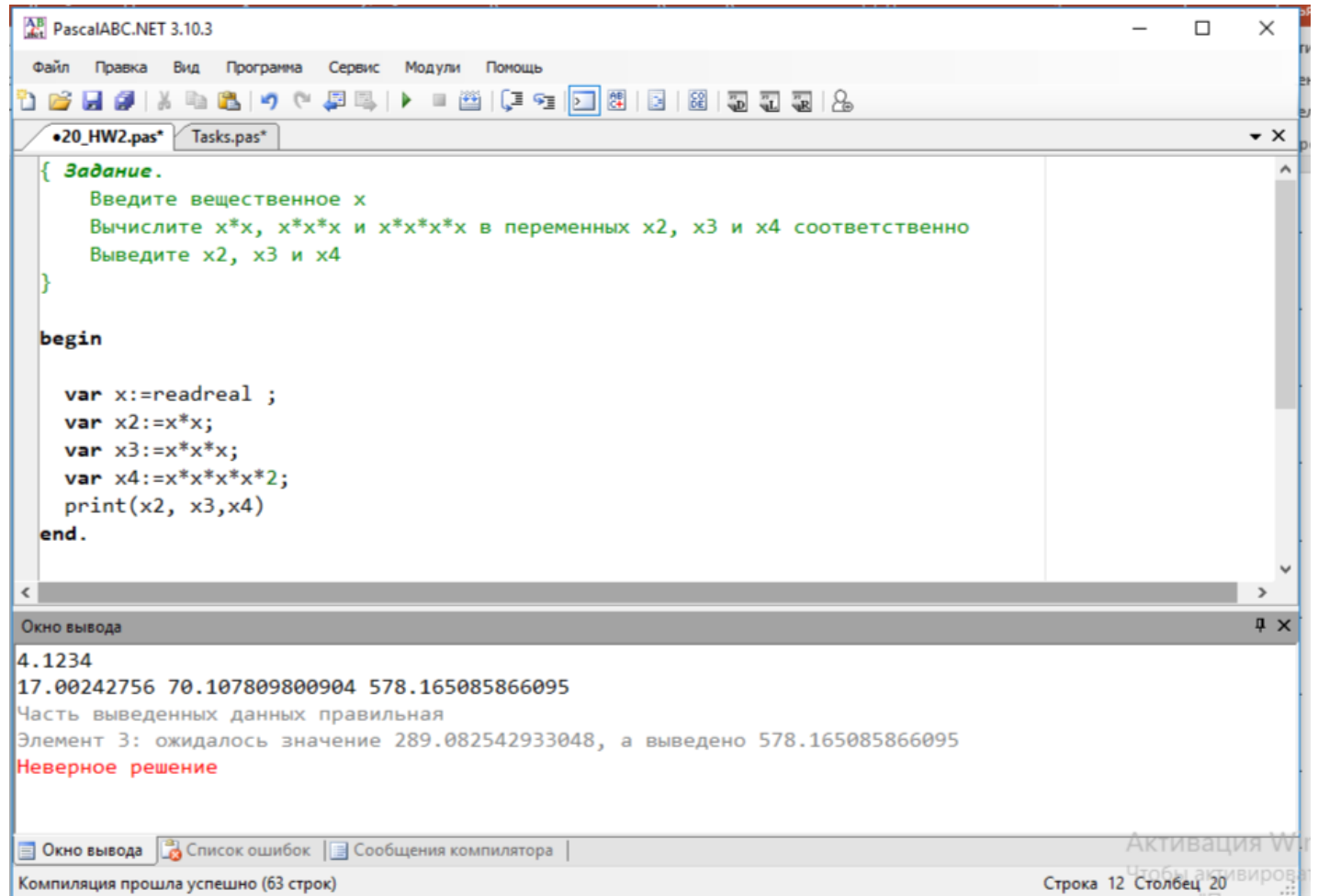
Below the code editor, the 'Окно вывода' (Output Window) is visible, showing the results of the program execution:

```
4.1234  
17.00242756 70.107809800904 289.082542933048  
Задание выполнено
```

The status bar at the bottom indicates 'Компиляция прошла успешно (63 строк)' (Compilation successful (63 lines)).

Разработка задания на некоторые простые проверки

Если всё хорошо, то надо убрать часть решения и проверить реакцию проверяющего алгоритма:



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main window displays a Pascal program named `Tasks.pas` with the following code:

```
{ Задание.  
  Введите вещественное x  
  Вычислите x*x, x*x*x и x*x*x*x в переменных x2, x3 и x4 соответственно  
  Выведите x2, x3 и x4  
}  
  
begin  
  
  var x:=readreal ;  
  var x2:=x*x;  
  var x3:=x*x*x;  
  var x4:=x*x*x*x*2;  
  print(x2, x3,x4)  
end.
```

The output window, titled "Окно вывода", shows the following output:

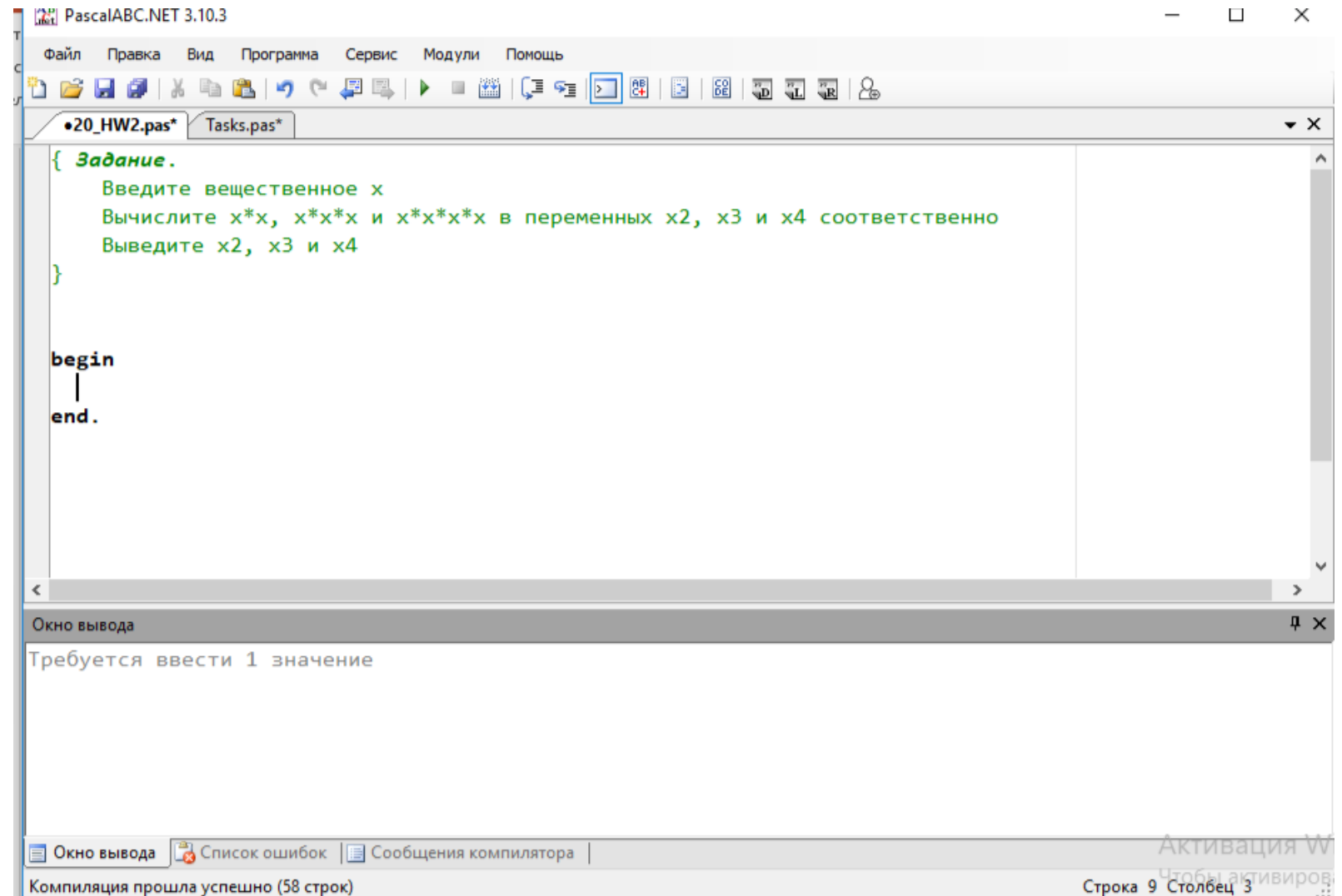
```
4.1234  
17.00242756 70.107809800904 578.165085866095  
Часть выведенных данных правильная  
Элемент 3: ожидалось значение 289.082542933048, а выведено 578.165085866095  
Неверное решение
```

The status bar at the bottom indicates "Компиляция прошла успешно (63 строк)" and "Строка 12 Столбец 20".

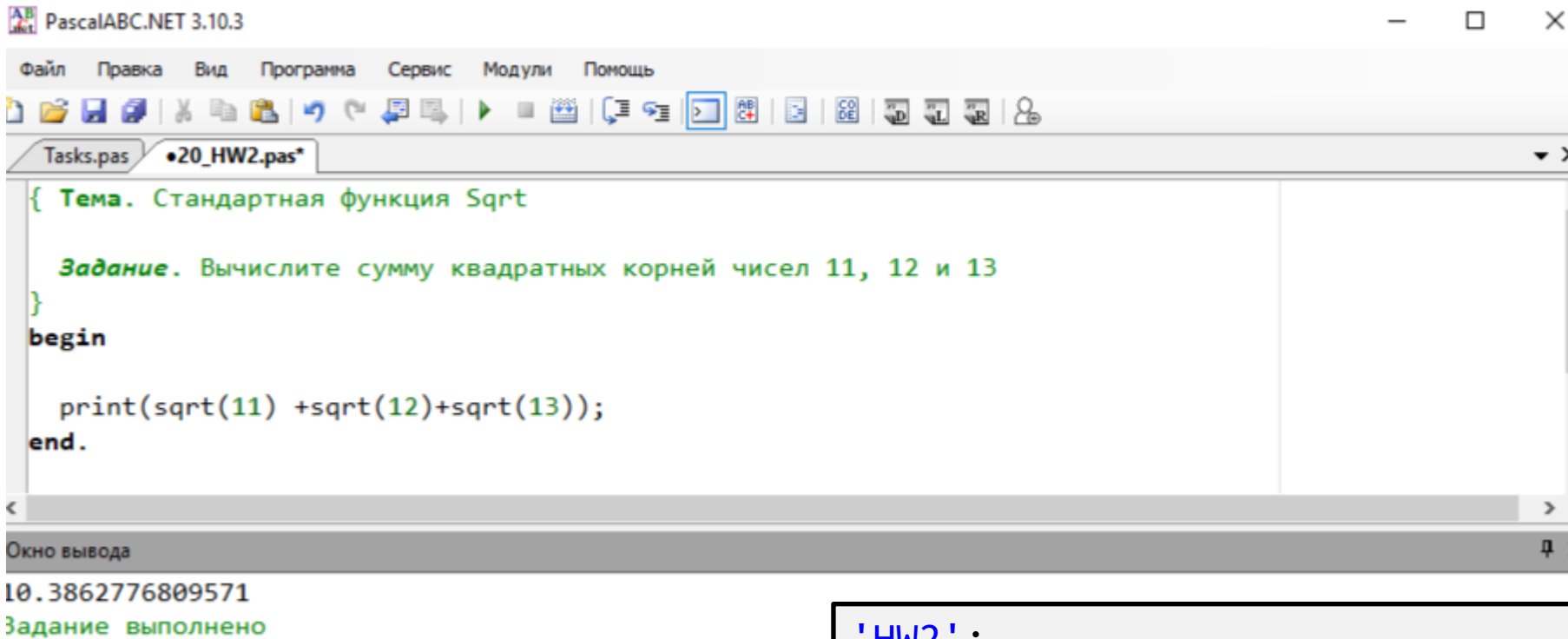
Разработка задания на некоторые простые проверки

И затем проверить, как работает пустая программа.

Если всё хорошо – задание готово!



Разработка задания, когда необходимо вывести одно число



The screenshot shows the PascalABC.NET 3.10.3 IDE. The main window displays a Pascal program in a file named `20_HW2.pas`. The program defines a task and calculates the sum of square roots of 11, 12, and 13. The output window at the bottom shows the result of the calculation and a confirmation message.

```
{ Тема. Стандартная функция Sqrt

Задание. Вычислите сумму квадратных корней чисел 11, 12 и 13
}
begin
    print(sqrt(11) +sqrt(12)+sqrt(13));
end.
```

Окно вывода

10.3862776809571
Задание выполнено

```
'HW2' :
begin
    FilterOnlyNumbers;
    CheckData(Input := Empty);
    CheckOutput(Sqrt(11)+Sqrt(12)+Sqrt(13));
end;
```

Разработка задания, когда необходимы дополнительные комментарии ученику

PascalABC.NET 3.10.3

Файл Правка Вид Программа Сервис Модули Помощь

Tasks.pas 20_HW2.pas* •00_DivMod01.pas*

```
{ Тема. Операции div и mod

a div b - целочисленное деление a на b
a mod b - остаток от целочисленного деления a на b

Запустите программу. Вычислите с ее помощью и запишите в тетрадь
результат целочисленного деления и остаток от деления
17 на 5
21 на 4
15 на 5
6 на 3
}
begin
  var a := 21;
  var b := 4;
  Println(a, 'div', b, '=', a div b);
  Println(a, 'mod', b, '=', a mod b);
end.
```

Окно вывода

```
21 div 4 = 5
21 mod 4 = 1
Перепишите результат в тетрадь
```

'DivMod01':

begin

FilterOnlyNumbers;

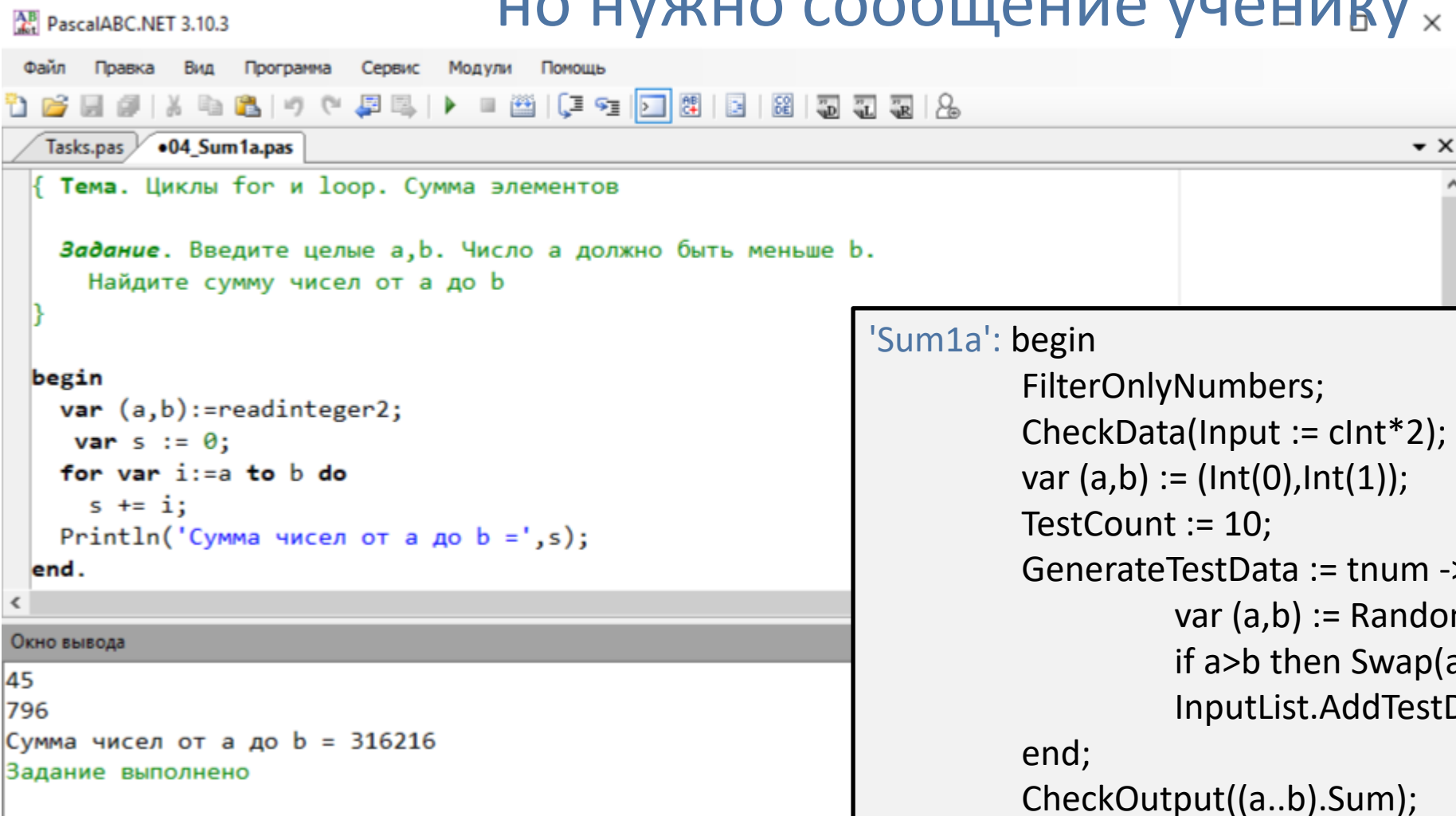
CheckData(Input := Empty);

// Просто поясняющее сообщение

ColoredMessage('Перепишите результат в тетрадь',
MsgColorGray);

end;

Разработка задания, когда не нужно ничего проверять, но нужно сообщение ученику



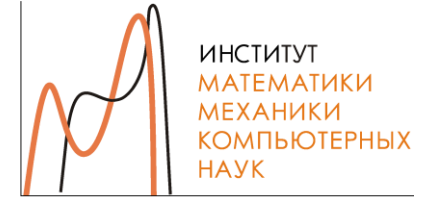
```
'Sum1a': begin
  FilterOnlyNumbers;
  CheckData(Input := clnt*2);
  var (a,b) := (Int(0),Int(1));
  TestCount := 10;
  GenerateTestData := tnum -> begin
    var (a,b) := Random2(1..20);
    if a>b then Swap(a,b);
    InputList.AddTestData(|a,b|);
  end;
  CheckOutput((a..b).Sum);
end;
```

Внесение результатов решения в базу данных

- Каждый запуск задания может сопровождаться внесением в базу данных или локальный файл информации об ученике, уроке, имени задания, времени запуска и результате решения
- Результат:
 - Solved** – задание решено
 - BadSolution** – задание не решено
 - PartialSolution** – частичное решение
 - NotUnderControl** – задание не проверяется
 - ErrFix** – задание на исправление ошибок
- Результаты, внесенные в базу данных, могут обрабатываться в режиме реального времени по данному уроку (вывод прогресса каждого ученика группы) и для создания сводной статистики
- В конце проверки каждого задания в **локальный файл db.txt** вносится информация (имя задания, время запуска, результат)
- Все папки учеников находятся на сетевом диске известной структуры, что делает возможной лёгкую обработку информации из всех локальных баз данных учеников
- Скрипт на PascalABC.NET сканирует эти файлы и формирует сводную информацию по ученикам за 1-2 секунды

Выводы

- Модуль невидимой автоматической проверки LightPT позволяет эффективно разрабатывать автоматическую проверку заданий **к каждому уроку**
- Ученик может решать задание **в свободном формате**: сам задавать типы данных, порядок ввода-вывода – проверяющий модуль способен правильно обрабатывать эти случаи
- Учитель может проверять результаты выполнения заданий в режиме реального времени (реализовано в виде текстовой локальной базы данных и в виде веб-системы)
- Система автоматической проверки может давать **интеллектуальные подсказки** в случае частичного решения
- Накапливаемая статистика может быть использована для автоматизированного анализа с целью улучшения качества заданий и изменения индивидуальных траекторий движения учеников
- С помощью модуля невидимой автоматической проверки можно составлять **контрольные работы**



*Южный федеральный университет,
Институт математики, механики и компьютерных наук*
tnbordyugova@sfedu.ru

