

Курс «CS101: Основы программирования»

для студентов направления
«Фундаментальная информатика и информационные технологии»

Николаев А. В.

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

Доклад на четвертой Всероссийской научно-методической конференции
«Использование системы программирования PascalABC.NET
в обучении программированию»
(Ростов-на-Дону, 29-30 марта 2023 г.)

Рассматриваемые темы

- Что изучают в курсе «Основы программирования» в университете?
- Почему PascalABC.NET?
- Частые ошибки студентов, привитые со школы

CS101. Основы программирования



```
graph TD; Root[CS101. Основы программирования] --> M1[Модуль 1]; Root --> M2[Модуль 2]; Root --> M3[Модуль 3]; M1 --- T1[Основные конструкции, циклы]; M2 --- T2[Подпрограммы, модули, библиотеки]; M3 --- T3[Классы, записи, строки];
```

Модуль 1

Основные конструкции,
циклы

Модуль 2

Подпрограммы, модули,
библиотеки

Модуль 3

Классы, записи, строки

Задачи для отработки базовых конструкций

Простые задачи на циклы и условия

Помимо обучения навыку правильного составления условий и использования циклов, делается акцент на подборе эффективных решений задачи.

8. {4 балла} [task-10-while.pas] Дано целое число $N > 1$. Найти наименьшее целое число K , при котором выполняется неравенство $3^K > N$.

Пример

```
N = 97 >>> K = 5  
N = 9 >>> K = 3  
N = 2 >>> K = 1
```

Замечание. Использовать стандартную функцию возведения в степень запрещено.

4. {2.5 балла} [task-04-if.pas] Даны три числа. Найти сумму двух наибольших из них.

Проверьте правильность вашей программы, приведите лог работы программы в форме комментария.

Замечание. Существует решение, в котором используется 2 операции сравнения.

Задачи для отработки базовых конструкций

Обработка цифр числа

Обучение составлению условий выхода из цикла, а также проверке своих программ на пограничных случаях

4. {2 балла} [task-04.pas] Дано целое число N . Вывести его цифры, начиная со старшего разряда. Например:

```
N = 1205 >>> 1, 2, 0, 5
N = -123 >>> 1, 2, 3
N = -111 >>> 1, 1, 1
N =    0 >>> 0
```

Указание 1. В этой задаче придётся использовать два цикла.

- В первом цикле нужно вычислить максимальную степень числа 10 ($\text{pow}10$), для которой выполнено: $N \text{ div } \text{pow}10 > 0$. Например, для числа 5256 это 1000, для числа 42 это 10 и т. п.
- Во втором цикле степень 10 следует использовать для извлечения нужной цифры, а также для её удаления из числа. Степень 10 уменьшается на каждом шаге этого цикла в 10 раз.

Указание 2. Чтобы программа корректно работала для чисел типа 10^k , условие второго цикла должно анализировать $\text{pow}10$, а не N .

Задачи для отработки базовых конструкций

Вычисление конечных сумм и рядов Тейлора

Рассматриваются проблемы при сравнении вещественных чисел, отрабатывается оптимизация количества вычислений для решения задачи.

4. {1.5 балла} [task-05.pas] Дано вещественное число A и целое число $N \geq 0$. Используя один цикл, найти сумму

$$1 + A + A^2 + A^3 + \dots + A^N.$$

Проверьте правильность вашей программы не менее чем на двух наборах данных. Обязательно проверьте случай $N = 0$.

Пример

```
A = 1.5, N = 10 >>> 170.9951171875
```

```
A = -3.7, N = 8 >>> 27651.64676491
```

4. [task-04-sqrt-sum.pas] Даны вещественные числа X ($|X| < 1$) и eps ($0 < eps < 1$). Найти значение выражения с точностью eps :

$$S = 1 + \frac{X}{2} - \frac{1 \cdot X^2}{2 \cdot 4} + \frac{1 \cdot 3 \cdot X^3}{2 \cdot 4 \cdot 6} - \frac{1 \cdot 3 \cdot 5 \cdot X^4}{2 \cdot 4 \cdot 6 \cdot 8} + \dots$$

Полученное число является приближенным значением функции $\sqrt{1+X}$ (sqrt(1 + x) в PascalABC.NET).

Задачи для отработки базовых конструкций

Рекуррентные соотношения

Развиваются навыки написания эффективных по количеству вычислений решений. Кроме того, особое внимание уделяется обработке пограничных случаев.

2. {1 балл} [task-02-fib-seq.pas] Дано целое число $N \geq 1$. Последовательность F_k **чисел Фибоначчи** (названа в честь [Леонардо Пизанского](#)) определяется следующим образом:

$$F_1 = 1, \quad F_2 = 1, \quad F_k = F_{k-2} + F_{k-1}, \quad k = 3, 4, \dots$$

Это целые числа:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \dots$$

Вывести элементы $F_1, F_2, \dots, F_N$.

Указание. Для вычисления чисел Фибоначчи разрешается использовать не более трёх переменных.

3. {1.5 балла} [task-03-seq2.pas] Дано целое число $N > 1$. Последовательность вещественных чисел A_k определяется следующим образом:

$$A_1 = 1, \quad A_2 = 2, \quad A_k = \frac{A_{k-2} + 2 \cdot A_{k-1}}{3}, \quad k = 3, 4, \dots$$

Вывести элементы $A_1, A_2, \dots, A_N$ и их среднее арифметическое.

А что с питонистами?

Средний балл за первый модуль у студентов, изучавших только *Python* в школе в среднем **на 21% ниже**, чем у студентов, изучавших *PascalABC.NET/C++*

Возникшие сложности:

- Полное отсутствие понимания системы типов
- Теряются при ошибках компиляции
- Долго не могут побороть привычку «значащих отступов»

А что с питонистами?

Python	PascalABC.NET
Простейшая программа	
<pre>print("Hello, World!")</pre>	<pre>begin Println('Hello, World!'); end.</pre>
	<pre>## Println('Hello, World!');</pre>
Комментарии в коде	
<pre># Однострочный комментарий ''' Многострочный Комментарий '''</pre>	<pre>// Однострочный комментарий { Многострочный комментарий }</pre>
Арифметические операции	
<pre>x + y x - y x * y x / y x // y x % y</pre>	<pre>x + y x - y x * y x / y x div y x mod y</pre>
Логические операции	
<pre>x and y</pre>	<pre>x and y</pre>

Изучение продвинутых тем

Подпрограммы и модули

- Полная/сокращённая форма модуля
 - Процедуры и функции, входные и входно-выходные параметры
 - Параметры по умолчанию
 - Функции как параметры и лямбда-выражения
-
- Основы работы с Git
 - Unit-тесты (NUnitABC)

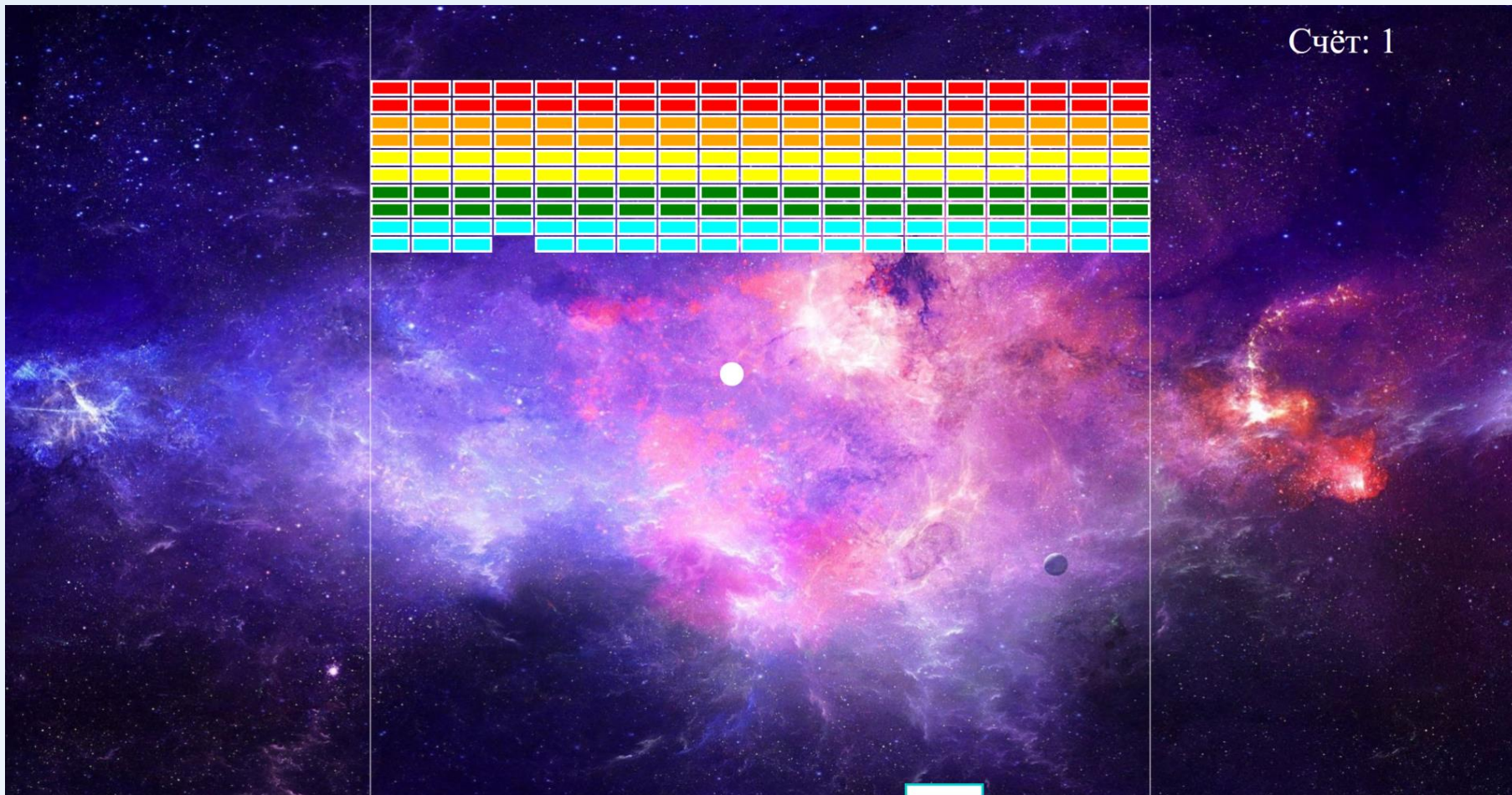
Изучение продвинутых тем

Графика (библиотека GraphWPF)



Изучение продвинутых тем

Графика (библиотека GraphWPF, WPFObjects)



Изучение продвинутых тем

Структуры данных

- Ссылочная модель данных
- Динамические массивы, методы работы с ними
- Передача массивов в качестве параметров
- Матрицы (двумерные массивы), методы работы с ними
- Срезы
- Последовательности
- Генерация последовательностей
- Методы последовательностей (LINQ: .Where, .Select и т.д.)
- Словари, множества, списки

Изучение продвинутых тем

Классы и структуры

- Ссылочная vs размерная модель
- Проектирование классов
- Статические методы классов
- Массивы объектов классов
- Строки

Почему не Python?

- Python – интерпретируемый язык, в то время как PascalABC.NET – компилируемый. Все семантические ошибки в PascalABC.NET отлавливаются ещё на этапе компиляции.
- Большинство IDE для Python перегружены инструментами для профессионалов, в которых начинающие программисты не нуждаются.
- Python – язык с динамической типизацией, в то время как во многих отраслях IT-индустрии преобладают языки со статической типизацией.

Намного проще перейти с языка со статической типизацией на Python, чем наоборот!

Почему не C++?

- При разработке на C++ необходимо работать с файлами проекта, в то время как в PascalABC.Net работа (в простейшем случае) ведётся с одним файлом.
- C++ крайне недружелюбен к начинающим программистам. Даже при решении простых задач можно допустить трудные для восприятия ошибки.
- C++ называют низкоуровневым языком, и действительно, сборка мусора, работа с указателями и памятью, и т.п. ложится на плечи разработчика. Но сможет ли выдержать этот груз начинающий программист?

Почему PascalABC.NET?

AB
C
.net

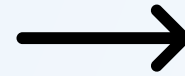
1 семестр



2 семестр



3 семестр



4-8 семестр

Типичные ошибки («плохой стиль»)

Работа с переменными в старом Паскаль-стиле

```
var
  i, n: integer;
  a, s: real;

begin
  read(a, n);
  s := 0.0;
  for i := 1 to n do
    s := s + a * i;
  writeln(s);
end
```

```
##
var a := ReadReal;
var n := ReadInteger;
var s: real := 0.0;

for var i := 1 to n do
  s += a * i;
println(s);
```

Типичные ошибки («плохой стиль»)

Устаревший подход к чтению с консоли

```
var
  a, n: integer;
  x, y, z: real;

begin
  read(n);

  write('Введите a: ');
  read(a);

  read(x, y, z);
end.
```

```
##
var n := ReadInteger;

var a := ReadInteger('Введите a: ');

var (x, y, z) := ReadReal3;
```

Типичные ошибки («плохой стиль»)

Устаревший подход к чтению с консоли

```
var
  a, n: integer;
  x, y, z: real;

begin
  read(n);

  write('Введите a: ');
  read(a);

  read(x, y, z);
end.
```

```
##
var n := ReadInteger;

var a := ReadInteger('Введите a: ');

var (x, y, z) := ReadReal3;
```

Типичные ошибки («плохой стиль»)

Использование if для печати boolean

```
##  
var a := ReadInteger();  
  
if a mod 2 = 0 then  
    Println(True)  
else  
    Println(False)
```

```
##  
var a := ReadInteger();  
  
Println(a mod 2 = 0)
```

Типичные ошибки («плохой стиль»)

Лишнее сравнение boolean в if

```
##  
var a := ReadInteger();  
  
if a.IsEven = true then  
    Println('Even');  
  
if a.Divs(10) = false then  
    Println('Не заканчивается на 0');
```

```
##  
var a := ReadInteger();  
  
if a.IsEven then  
    Println('Even');  
  
if not a.Divs(10) then  
    Println('Не заканчивается на 0');
```

Типичные ошибки («плохой стиль»)

Игнорирование операторов +=, *=, ...

```
##  
var a := ReadInteger();  
var (sum, prod) := (0, 1);  
  
while a <> 0 do  
begin  
    sum := sum + a mod 10;  
    prod := prod + a mod 10;  
    a := a div 10;  
end;  
  
Print(sum, prod);
```

```
##  
var a := ReadInteger();  
var (sum, prod) := (0, 1);  
  
while a <> 0 do  
begin  
    sum += a mod 10;  
    prod *= a mod 10;  
    a := a div 10;  
end;  
  
Print(sum, prod);
```

Николаев А. В.

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

andn@sfedu.ru