

## Самостоятельная работа №3

### Задание 1: Вычислить

**Задача:** Вычислить интеграл табличной функции используя формулу Ньютона-Лейбница. Результат записать.

**Табличная функция:**  $f(x) = x^2$

**Интеграл:**  $\int_1^{10} x^2 dx = F(10) - F(1) = \frac{10^3}{3} - \frac{1^3}{3} = \frac{999}{3} = 333$

### Задание 2.1: Метод левых сторон прямоугольников

**Задача:** После составления кода программы на вычисление определенного интеграла одним из численных методов, протестируйте правильность ее работы на выбранной вами в п.1 табличной подынтегральной функции. Результат представить в отчете по самостоятельной работе.

#### Описание переменных

Переменная	Тип	Суть
a	real	Нижний предел интегрирования
b	real	Верхний предел интегрирования
x	real	Аргумент функции
f	real	Значение подынтегральной функции
integral	real	Искомый интеграл
h	real	Шаг
n	integer	Количество разбиений

## Код программы

```
main.pas
1  program SR3_1;
2  var
3      a, b, x, f, h, integral: real;
4      n: integer;
5  begin
6      a := 1;
7      b := 10;
8      n := 1;
9
10     while n <> 10000 do
11     begin
12         n := n * 10;
13         h := (b - a) / n;
14         integral := 0;
15
16         x := a;
17         while x <= b do begin
18             f := sqr(x);
19             x := x + h;
20             integral := integral + f;
21         end;
22
23         integral := integral * h;
24         writeln('Интеграл при ', n, ' разбиений = ', integral:0:7);
25     end;
26 end.
```

input

Compiled Successfully. memory: 1556 time: 0 exit code: 0

```
Интеграл при 10 разбиений = 289.6650000
Интеграл при 100 разбиений = 337.5571500
Интеграл при 1000 разбиений = 332.5546215
Интеграл при 10000 разбиений = 333.0454512
```

### Задание 3.1: Метод левых сторон прямоугольников

**Задача:** Сравнить результаты нахождения интеграла табличной функции «вручную» (через первообразную, по формуле Ньютона-Лейбница) и найденный в результате программной реализации.

**Сравнение результата работы программы и расчета вручную:** Как можно видеть из результатов работы программы, по сравнению с формулой Ньютона-Лейбница присутствует некоторая погрешность. Погрешность уменьшается при увеличении числа разбиений и уже после 1000 разбиений достигает значения менее чем в единицу. Однако, увеличение числа разбиений приводит к увеличению числа итераций цикла, что означает большее потребление ресурсов машиной.

## Задание 2.2: Метод правых сторон прямоугольников

**Задача:** После составления кода программы на вычисление определенного интеграла одним из численных методов, протестируйте правильность ее работы на выбранной вами в п.1 табличной подынтегральной функции. Результат представить в отчете по самостоятельной работе.

### Описание переменных

Переменная	Тип	Суть
a	real	Нижний предел интегрирования
b	real	Верхний предел интегрирования
x	real	Аргумент функции
f	real	Значение подынтегральной функции
integral	real	Искомый интеграл
h	real	Шаг
n	integer	Количество разбиений

## Код программы

```
main.pas
1  program SR3_2;
2  var
3      a, b, x, f, h, integral: real;
4      n: integer;
5  begin
6      a := 1;
7      b := 10;
8      n := 1;
9
10     while n <= 10000 do
11     begin
12         n := n * 10;
13         h := (b - a) / n;
14         integral := 0;
15
16         x := a;
17         while x < b do begin
18             x := x + h;
19             f := sqr(x);
20
21             integral := integral + f;
22         end;
23
24         integral := integral * h;
25         writeln('Интеграл при ', n, ' разбиений = ', integral:0:7);
26     end;
27 end.
```

input

Compiled Successfully. memory: 1580 time: 0 exit code: 0

```
Интеграл при 10 разбиений = 378.7650000
Интеграл при 100 разбиений = 346.6298790
Интеграл при 1000 разбиений = 333.4456215
Интеграл при 10000 разбиений = 333.1345674
```

### Задание 3.2: Метод правых сторон прямоугольников

**Задача:** Сравнить результаты нахождения интеграла табличной функции «вручную» (через первообразную, по формуле Ньютона-Лейбница) и найденный в результате программной реализации.

**Сравнение результата работы программы и расчета вручную:** Как можно видеть из результатов работы программы, по сравнению с формулой Ньютона-Лейбница присутствует некоторая погрешность. Погрешность уменьшается при увеличении числа разбиений и уже после 1000 разбиений достигает значения менее чем в единицу. Однако, увеличение числа разбиений приводит к увеличению числа итераций цикла, что означает большее потребление ресурсов машиной.

### Задание 2.3: Метод трапеций

**Задача:** После составления кода программы на вычисление определенного интеграла одним из численных методов, протестируйте правильность ее работы на выбранной вами в п.1 табличной подынтегральной функции. Результат представить в отчете по самостоятельной работе.

#### Описание переменных

Переменная	Тип	Суть
a	real	Нижний предел интегрирования
b	real	Верхний предел интегрирования
x	real	Аргумент функции
f	real	Среднее значение подынтегральной функции
f1	real	Значение подынтегральной функции в точке X
f2	real	Значение подынтегральной функции в точке X + h
integral	real	Искомый интеграл
h	real	Шаг
n	integer	Количество разбиений

## Код программы

```
main.pas
1  program SR3_3;
2  var
3      a, b, x, f, f1, f2, h, integral: real;
4      n: integer;
5  begin
6      a := 1;
7      b := 10;
8      n := 1;
9
10     while n <= 10000 do
11     begin
12         n := n * 10;
13         h := (b - a) / n;
14         integral := 0;
15
16         x := a;
17         while x < b do begin
18             f1 := sqr(x);
19             x := x + h;
20             f2 := sqr(x);
21             f := (f1 + f2) / 2;
22             integral := integral + f;
23         end;
24
25         integral := integral * h;
26         writeln('Интеграл при ', n, ' разбиений = ', integral:0:7);
27     end;
28 end.
```

input

Compiled Successfully. memory: 1504 time: 0 exit code: 0

```
Интеграл при 10 разбиений = 334.2150000
Интеграл при 100 разбиений = 342.0935145
Интеграл при 1000 разбиений = 333.0001215
Интеграл при 10000 разбиений = 333.0900093
```

### Задание 3.3: Метод трапеций

**Задача:** Сравнить результаты нахождения интеграла табличной функции «вручную» (через первообразную, по формуле Ньютона-Лейбница) и найденный в результате программной реализации.

**Сравнение результата работы программы и расчета вручную:** Как можно видеть из результатов работы программы, по сравнению с формулой Ньютона-Лейбница присутствует некоторая погрешность. Погрешность уменьшается при увеличении числа разбиений и уже после 1000 разбиений достигает значения менее чем в единицу. Однако, увеличение числа разбиений приводит к увеличению числа итераций цикла, что означает большее потребление ресурсов машиной.

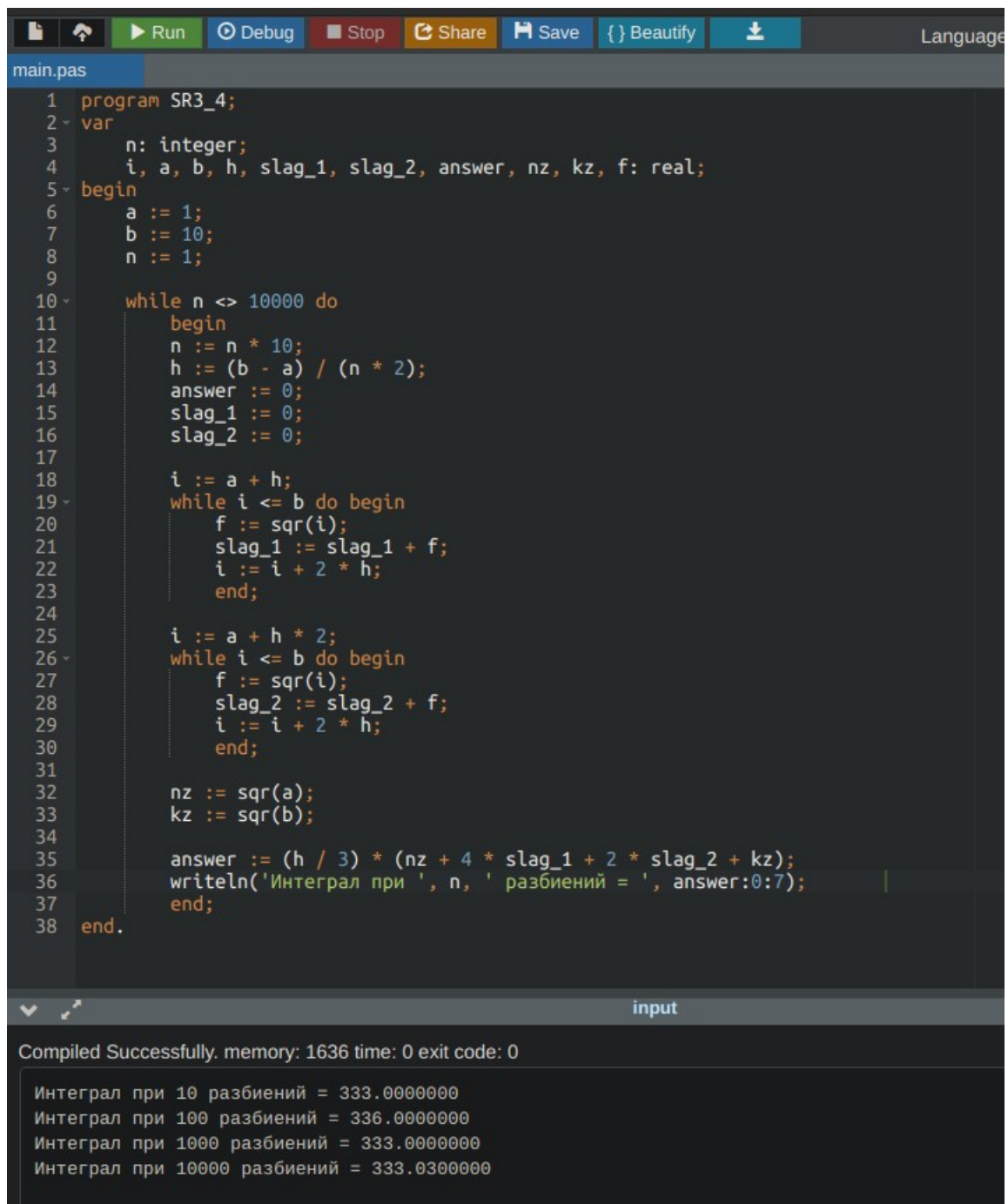
## Задание 2.4: Метод парабол

**Задача:** После составления кода программы на вычисление определенного интеграла одним из численных методов, протестируйте правильность ее работы на выбранной вами в п.1 табличной подынтегральной функции. Результат представить в отчете по самостоятельной работе.

### Описание переменных

Переменная	Тип	Суть
a	real	Нижняя граница интеграла
b	real	Верхняя граница интеграла
n	integer	Количество разбиений
h	real	Шаг итерации
i	real	Счетчик цикла
answer	real	Значение интеграла
slag_1	real	Слагаемое 1 в скобках
slag_2	real	Слагаемое 2 в скобках
nz	real	Значение ф-ии в нижней границе интегрирования
kz	real	Значение ф-ии в верхней границе интегрирования
f	real	Значение подынтегральной функции

## Код программы



```
1 program SR3_4;
2 var
3     n: integer;
4     i, a, b, h, slag_1, slag_2, answer, nz, kz, f: real;
5 begin
6     a := 1;
7     b := 10;
8     n := 1;
9
10    while n <> 10000 do
11    begin
12        n := n * 10;
13        h := (b - a) / (n * 2);
14        answer := 0;
15        slag_1 := 0;
16        slag_2 := 0;
17
18        i := a + h;
19        while i <= b do begin
20            f := sqr(i);
21            slag_1 := slag_1 + f;
22            i := i + 2 * h;
23        end;
24
25        i := a + h * 2;
26        while i <= b do begin
27            f := sqr(i);
28            slag_2 := slag_2 + f;
29            i := i + 2 * h;
30        end;
31
32        nz := sqr(a);
33        kz := sqr(b);
34
35        answer := (h / 3) * (nz + 4 * slag_1 + 2 * slag_2 + kz);
36        writeln('Интеграл при ', n, ' разбиений = ', answer:0:7);
37    end;
38 end.
```

input

Compiled Successfully. memory: 1636 time: 0 exit code: 0

```
Интеграл при 10 разбиений = 333.0000000
Интеграл при 100 разбиений = 336.0000000
Интеграл при 1000 разбиений = 333.0000000
Интеграл при 10000 разбиений = 333.0300000
```



### **Задание 3.4: Метод парабол**

**Задача:** Сравнить результаты нахождения интеграла табличной функции «вручную» (через первообразную, по формуле Ньютона-Лейбница) и найденный в результате программной реализации.

**Сравнение результата работы программы и расчета вручную:** Этот метод оказался наиболее точным из исследованных, и уже при 10 разбиений результат вычислений машины совпал с результатом ручных расчетов по формуле Ньютона-Лейбница, однако это может быть вызвано природой выбранной подынтегральной функции, чей график является параболой.