

Лабораторная работа №15 по дисциплине "Программирование"

ТЕМА: Алгоритмы с использованием случайных величин. Рекурсивные алгоритмы.

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПЕК

1. В Вашей папке создайте папку с именем П_лр15, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.
2. При выполнении задания создавайте в папке П_лр15 для каждой программы папку с именем, по смыслу соответствующим действиям, выполняемым составленной программой, и сохраняйте все файлы, имеющие отношение к этой программе, в этой созданной папке (имена сохраняемых файлов должны отличаться от имен, присваиваемых по умолчанию, и должны по смыслу соответствовать действиям, выполняемым составленной программой).
3. ПРИ ВЫПОЛНЕНИИ СЛЕДУЮЩИХ ЛАБОРАТОРНЫХ РАБОТ ДЕЙСТВУЙТЕ АНАЛОГИЧНО ПРЕДЫДУЩЕМУ.

АЛГОРИТМЫ С ИСПОЛЬЗОВАНИЕМ СЛУЧАЙНЫХ ВЕЛИЧИН

4. При реализации алгоритмов с использованием случайных величин могут быть использованы следующие функции и процедуры генерации случайных чисел:

Процедура или функция с именем и примером перечня аргументов	Описание и примеры использования
Функция RandG(m, sq)	<p>Генерирует случайные числа, распределенные по нормальному закону с математическим ожиданием m и средним квадратическим отклонением sq.</p> <p>Например, оператор for i:=1 to N do A[i]:=RandG(5,0.5); заполняет N элементов массива A случайными числами, распределенными по нормальному закону с математическим ожиданием 5 и средним квадратическим отклонением 0.5.</p>
Функция Random(R)	<p>Генерирует равномерно распределенные действительные или целые случайные числа. Если необязательный целочисленный параметр R не задан, то функция возвращает случайные действительные числа, равномерно распределенные в диапазоне $0 \leq X < 1$ или, по другому обозначению, в диапазоне [0,1).</p> <p>Например, оператор for i:=1 to N do A[i]:=Random; заполняет N элементов массива A действительными случайными числами, равномерно распределенными в диапазоне [0,1).</p> <p>Другой "статистический" пример: после выполнения операторов NL:=0; for i:=1 to N do if Random<L then NL:=NL+1; для значения $0 < L < 1$ можно убедиться, что при достаточно большом значении N отношение NL к N будет близко к значению L.</p> <p>Если необходимы действительные случайные числа, равномерно распределенные в диапазоне $X_1 \leq X < X_2$, то соответствующий пример с заполнением элементов массива может выглядеть следующим образом: for i:=1 to N do A[i]:=X1+(X2-X1)*Random;</p> <p>Если задан целочисленный параметр R, то функция возвращает случайные целые числа, равномерно распределенные в диапазоне $0 \leq X < R$ или, по другому обозначению, в диапазоне [0,R).</p> <p>Например, оператор for i:=1 to N do A[i]:=Random(R+1); заполняет N элементов массива A целыми случайными числами, равномерно распределенными в диапазоне $0 \leq X \leq R$ или, по другому обозначению, в диапазоне [0,R).</p> <p>Если необходимы целые случайные числа, равномерно распределенные в диапазоне $R_1 \leq X \leq R_1+R$, то аналогичный пример может выглядеть следующим образом: for i:=1 to N do A[i]:=R1+Random(R+1);</p>
Процедура Randomize	<p>Задаёт случайное начальное значение для функций RandG или Random (иначе говоря, рандомизирует эти функции).</p> <p>Если перед первым обращением к любой из указанных функций не использовать эту процедуру, то при каждом запуске программы функции RandG и Random будут генерировать одинаковые последовательности случайных чисел. При использовании же этой процедуры при каждом запуске программы последовательности случайных чисел будут разными.</p> <p>Например: Randomize;</p> <p>for i:=1 to N do A[i]:=Random;</p> <p>Имеется также целочисленная переменная RandSeed, хранящая начальное число, из которого формируется последовательность генерируемых случайных чисел. При повторном задании этой переменной значения, которое она имела перед генерацией, можно повторить последовательность генерируемых случайных (точнее, квазислучайных) чисел.</p>

РЕКУРСИВНЫЕ АЛГОРИТМЫ

5. Рекурсивный алгоритм предполагает использование этого же алгоритма при своей реализации. При реализации рекурсивного алгоритма используется рекурсивная подпрограмма (процедура или функция), в которой содержится обращение к этой же подпрограмме.
- При обращении подпрограммы к самой себе, как и при обращении к любой другой подпрограмме, используется стек для записи информации, необходимой для работы подпрограммы и для последующего выхода из нее и возврата в вызывающую программу. При повторных вызовах этот процесс повторяется.
- Для завершения вычислений каждая рекурсивная подпрограмма должна содержать нерекурсивную часть алгоритма, реализующую возврат в вызывающую программу и вызывающую соответствующее освобождение стека.
- Например, для вычисления суммы первых N чисел натурального ряда может быть использовано рекуррентное соотношение вида $S_N = S_{N-1} + N$ при условии, что $S_1 = 1$. Описание соответствующей рекурсивной функции может выглядеть следующим образом:
- ```
Function sumN(N:integer):integer;
begin
 if n=1 then sumN:=1 //Нерекурсивная часть
 else sumN:=sumN(N-1)+N; //Рекурсивная часть
end;
```

## СОЗДАНИЕ ПРОГРАММ

ВНИМАНИЕ! Все создаваемые здесь программы должны быть "зациклены", то есть после вывода результатов следует переходить снова на ввод данных.

### ПРОГРАММА 15-1

6. Создайте консольное приложение, реализующее приведенный выше "статистический" пример использования функции Random с вводом действительного значения  $L$ , целочисленного значения  $N$  и выводом отношения  $NL$  к  $N$  с 5-ю знаками после десятичного разделителя (данное приложение фактически реализует статистический метод с числом испытаний  $N$  "определения" длины отрезка, точки которого расположены в промежутке от 0 до  $L \leq 1$  внутри единичного отрезка).

### ПРОГРАММА 15-2

7. Создайте консольное приложение, выполняющее действия ПРОГРАММЫ 15-1 для диапазона действительных случайных чисел  $[X_1, X_2]$ , где значение  $X_1$  равно номеру Вашего компьютера, значение  $X_2$  равно 50.

### ПРОГРАММА 15-3

8. Создайте консольное приложение, выполняющее действия ПРОГРАММЫ 15-2 для диапазона целых случайных чисел  $[X_1, X_2]$ , где значение  $X_1$  равно номеру Вашего компьютера, значение  $X_2$  равно 50 (с вводом целого значения  $L$ ).

### ПРОГРАММА 15-4

9. Создайте консольное приложение, реализующее, по аналогии с ПРОГРАММОЙ 15-1, статистический метод определения площади прямоугольника со сторонами  $L_x \leq 1$  и  $L_y \leq 1$  внутри единичного квадрата с вводом действительных значений  $L_x$  и  $L_y$ ,

целочисленного значения числа испытаний  $N$  и выводом результата с 5-ю знаками после десятичного разделителя (можно считать, что левый нижний угол квадрата находится в начале координат и что стороны прямоугольника внутри квадрата откладываются от начала координат).

ПРОГРАММА 15-5

-----

10. Создайте консольное приложение, реализующее, по аналогии с ПРОГРАММОЙ 15-4, статистический метод определения площади под кривой с уравнением  $y=x^2$  внутри единичного квадрата с вводом целочисленного значения числа испытаний  $N$  и выводом результата с 5-ю знаками после десятичного разделителя (можно считать, что левый нижний угол квадрата находится в начале координат).

ПРОГРАММА 15-6

-----

11. Создайте консольное приложение, использующее рекурсивную функцию для вычисления значения факториала целого неотрицательного числа. Для вычисления факториала числа  $N$  (обозначается как  $N!$ ) может быть использовано рекуррентное соотношение вида  $N! = N \cdot (N-1)!$  при условии, что  $0! = 1$ ,  $1! = 1$ .  
В приложении должен быть предусмотрен ввод целого числа  $N$ , вывод значения  $N!$ , а также вывод сообщения, если введенное значение  $N < 0$ .

ПРОГРАММА 15-7

-----

12. Создайте консольное приложение, использующее рекурсивную функцию для вычисления значения  $N$ -го числа Фибоначчи. Для вычисления этого числа (обозначим его как  $F_N$ ) может быть использовано рекуррентное соотношение вида  $F_N = F_{N-1} + F_{N-2}$  при условии, что  $F_1 = F_2 = 1$ .  
В приложении должен быть предусмотрен ввод целого числа  $N$ , вывод значения  $F_N$ , а также вывод сообщения, если введенное значение  $N < 0$ .

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ

-----

13. Продемонстрируйте тексты созданных программ, а также их выполнение в среде программирования и независимо от нее.
-