**Построение диаграмм**

Пусть имеется последовательность положительных действительных чисел *a*1, *a*2, ..., *an*, обозначающая результаты каких-либо измерений (например, высоты вершин гор над уровнем моря, площади государств, средние оценки учеников класса и т.д.). Требуется построить визуализированное представление этой последовательности с целью сравнения полученных результатов. В таких случаях используют диаграммы.

**1. Круговые диаграммы**

В круговой диаграмме каждому элементу последовательности соответствует сектор, градусная мера которого пропорциональна величине элемента.

Для построения круговой диаграммы необходимо просуммировать все элементы последовательности, после чего найти отношения каждого из элементов к полученной сумме (так будет вычислено, какую часть круга нужно поставить в соответствие данной величине, — т.е. рассчитываются доли круга, приходящиеся на данную величину, если весь круг принять равным 1). Все эти расчеты можно представить формулами . Затем эти относительные величины переводятся в градусы: , после чего можно приступать к построению диаграммы.

Алгоритм в этом случае будет следующим:

1. вычислить сумму элементов последовательности;
2. найти величину сектора, соответствующего каждой величине;
3. построить все секторы в графическом режиме (в результате должен получиться полный круг). Желательно каждый сектор строить своим цветом, или использовать разную штриховку, если секторы одноцветные.

Программа построения круговой диаграммы по этому алгоритму представлена ниже:

{Круговая диаграмма (с) А.П. Шестаков, 2001}

program Kr\_D;

Uses Graph;

Var a, S : Real; I : Byte; G, M : Integer;

 Xc, Yc, R : Integer; {координаты центра круга и его радиус}

 F : Text; {файл содержит данные для построения диаграммы}

 Alpha : Integer; {угол, соответствующий очередной величине}

 SAngle : Integer; Stroka : String;

Begin

 Assign(F, '1.dat'); Reset(F);

 S := 0; {сумма элементов последовательности}

 While Not Eof(F) Do

 begin Readln(F, a); S := S + a end;

 reset(f); G := detect; M := 0;

 initgraph(G, M, ''); Xc := GetMaxX Div 2; Yc := GetMaxY Div 2;

 R := 100; SAngle := 0; i := 1;

 While Not Eof(f) Do begin

 Readln(F, a); Alpha := round(A / S \* 360); {вычисление угла}

 setcolor(i mod 16 + 1); setfillstyle(1, i mod 16 + 1);

{построение сектора, соответствующего величине}

 sector(Xc, Yc, SAngle, SAngle + Alpha, R, R);

 SAngle := SAngle + Alpha; i:= i + 1;

 {укажем, какому цвету какая величина соответствует}

 bar(Xc+200, Yc-250+(i-1)\*20, Xc+220, Yc-250+(i-1)\*20+15);

 str(a:8:2, stroka);

 outtextxy(Xc + 230, Yc — 250 + 5 + (i — 1) \* 20, stroka) end;

 readln; close(F); closegraph End.

Результат работы программы для указанного на рисунке набора чисел:



**2. Столбчатые диаграммы**

Для построения диаграммы выделим на экране прямоугольную область с координатами соответственно верхнего левого угла (Xlv, Ylv) и правого нижнего (Xpn, Ypn). Высота столбца диаграммы, соответствующего максимальному элементу последовательности, будет совпадать с высотой прямоугольника. Ширина столбца будет зависеть от количества элементов последовательности: чем больше компонент, тем меньшей будет ширина. Таким образом, для построения диаграммы нужно определить количество компонентов последовательности и максимальный элемент последовательности. Высота *vi* очередного столбца диаграммы на экране будет определяться формулой где xmax — максимальный элемент последовательности, *xi* — очередной элемент последовательности.

Алгоритм построения диаграммы следующий:

1. определить количество элементов последовательности и её максимальный элемент;
2. согласно указанной формуле построить столбцы диаграммы. Их ширина на экране может быть рассчитана по формуле где *n* — количество элементов последовательности.

Программа построения столбчатой диаграммы по этому алгоритму представлена ниже:

program Stol\_D;

Uses Graph;

Var a, xmax : Real; I, n : Byte; G, M : Integer;

 F : Text; {файл содержит данные для построения диаграммы}

 Stroka : String;

 Xlv, Ylv, Xpn, Ypn : Integer; {координаты окна вывода диаграммы}

Begin

 Assign(F, '1.dat'); Reset(F);

 if not eof(f) then begin readln(f, xmax); n:= 1 end else n := 0;

 While Not Eof(F) Do

 begin Readln(F, a); if a > xmax then xmax := a; n := n + 1 end;

 reset(f); G := detect; M := 0;

 initgraph(G, M, ''); Xlv := 50; Ylv := 50; Xpn:= GetMaxX-100; Ypn:= GetMaxY-50;

 i:= 0; {номер столбца}

 While Not Eof(f) Do

 begin

 Readln(F, a);

 setcolor(i mod 16 + 1); setfillstyle(1, i mod 16 + 1);

 {очередной столбец}

 bar(round(Xlv+i\*(Xpn-Xlv)/n), Ypn,

 round(Xlv+(i+1)\*(Xpn-Xlv)/n), round(Ypn-(Ypn-Ylv)/xmax\*a));

 i:= i + 1;

 {укажем, какому цвету какая величина соответствует}

 bar(getMaxx-70, 50+(i-1)\*20, getMaxx-50, 50+(i-1)\*20+15);

 str(a:8:2, stroka);

 outtextxy(getMaxx-40, 50+(i-1)\*20+8, stroka);

 end;

 readln; close(F); closegraph

End.

Результат работы программы для указанного на рисунке набора чисел:



Для лучшего восприятия диаграммы было бы целесообразно построить вертикальную ось с разметкой по ней, что в данной программе отсутствует.

**3. Линейные диаграммы**

При построении линейных диаграмм каждой величине соответствует точка, расположенная на определённой высоте относительно начала отсчёта (высота рассчитывается так же, как и при построении столбчатых диаграмм), все точки соединяются линиями. В результате получается ломаная. Такого рода диаграммы чаще всего строя в тех случаях, когда необходимо визуализировать динамику изменения величин.

Программа аналогична программе построения столбчатых диаграмм и приведена ниже.

{Линейная диаграмма (с) А.П. Шестаков, 2001}

program Stol\_D;

Uses Graph;

Var a, xmax : Real; I, n : Byte; G, M : Integer;

 F : Text; {файл содержит данные для построения диаграммы}

 Stroka : String; Yn, Yk : Integer;

 Xlv, Ylv, Xpn, Ypn : Integer; {координаты окна вывода диаграммы}

Begin

 Assign(F, '1.dat'); Reset(F);

 if not eof(f) then begin readln(f, xmax); n:= 1 end else n := 0;

 While Not Eof(F) Do

 begin Readln(F, a); if a > xmax then xmax := a; n := n + 1 end;

 reset(f); G := detect; M := 0;

 initgraph(G, M, ''); Xlv := 50; Ylv := 50; Xpn:= GetMaxX-100; Ypn:= GetMaxY-50;

 line(xlv, ylv, xlv, ypn); line(xlv, ypn, xpn, ypn);

 i:= 0; {номер точки}

 readln(f, a);

 Yn := round(Ypn-(Ypn-Ylv)/xmax\*a);

 str(a:5:1, stroka);

 outtextxy(round(Xlv+i\*(Xpn-Xlv)/n)-20, Ypn+20, stroka);

 While Not Eof(f) Do

 begin

 setcolor(i mod 16 + 1); setfillstyle(1, i mod 16 + 1);

 {укажем, какому цвету какая величина соответствует}

 Readln(F, a);

 Yk := round(Ypn-(Ypn-Ylv)/xmax\*a);

 {очередная линия}

 line(round(Xlv+i\*(Xpn-Xlv)/n), Yn,

 round(Xlv+(i+1)\*(Xpn-Xlv)/n), Yk);

 i:= i + 1;

 str(a:5:1, stroka);

 outtextxy(round(Xlv+i\*(Xpn-Xlv)/n)-20, Ypn+20, stroka);

 Yn := Yk; {запоминаем положение очередной точки}

 end;

 readln; close(F); closegraph

End.

Результат работы программы для указанного на рисунке набора чисел:



Все представленные здесь программы могут быть объединены в одну программу с общим меню, где пользователю предоставляется возможность выбрать вид диаграммы.

**Примеры построения диаграмм в MS Excel**

Имеется некоторый набор числовых данных, который требуется визуализировать.

 По мнению аналитиков, население мира в последующие годы будет расти и к 2030 году может выглядеть таким образом:

1-е место: Индия – 1 449 078 000 чел.

2-е место: Китай – 1 420 296 000 чел.

3-е место: США – 360 894 000 чел.

4-е место: Индонезия – 270 844 000 чел.

5-е место: Пакистан – 246 322 000 чел.

6-е место: Бразилия – 235 505 000 чел.

7-е место: Бангладеш – 205 641 000 чел.

8-е место: Нигерия – 204 465 000 чел.

9-е место: Мексика – 133 221 000 чел.

10-е место: Эфиопия – 128 979 000 чел.

Занесем данные в таблицу:



Построим **круговую диаграмму**. Для этого воспользуемся **Мастером диаграмм**. 

Выберем **круговую диаграмму**.



Далее укажем диапазон данных, по которым будет построена диаграмма. Для этого достаточно выделить данные в столбце (вместе с данными выделим также названия стран, чтобы на диаграмме можно было указать, что к какой стране относится).



Укажем заголовки, легенду и подписи данных



Сохраним диаграмму на этом же листе, что и данные



Аналогичным образом можно построить линейную или столбчатую (гистограмму) диаграммы

