**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**

**НАБЕРЕЖНОЧЕЛНИНСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ**

**МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**

**КАФЕДРА МАТЕМАТИКИ И МЕТОДИКИ ЕЕ ПРЕПОДАВАНИЯ**

**АЛГОРИТМЫ С МГНОГОЧЛЕНАМИ**

**/дипломная работа/**

**Набережные Челны**

**2006 год**

**Содержание**

[Введение](#_Toc137620045)

[1. Многочлены](#_Toc137620046)

[2. Деление многочленов](#_Toc137620047)

[2.1. Делимость многочленов. Свойства делимости](#_Toc137620048)

[2.2. Деление многочленов с остатком](#_Toc137620049)

[2.3. Наибольший общий делитель многочленов](#_Toc137620050)

[2.4. Алгоритм Евклида](#_Toc137620051)

[3. Кратные корни](#_Toc137620052)

[4. Производная от многочлена](#_Toc137620053)

[5. Кратные множители](#_Toc137620054)

[5.1. Выделение кратных множителей](#_Toc137620055)

[Заключение](#_Toc137620056)

[Список использованной литературы](#_Toc137620057)

**Введение**

Тема моей дипломной работы: «Алгоритмы с многочленами».

Целью данной работы является изучение многочленов, алгоритмов с ними, рассмотрение возможностей составления различных программ. Для достижения поставленной цели необходимо рассмотреть следующие вопросы:

– делимость многочленов;

– деление многочленов с остатком;

– наибольший общий делитель, алгоритм Евклида;

– кратные корни;

– кратные множители, выделение кратных множителей;

– производные от многочленов.

Для выполнения дипломной работы я поставила следующие задачи:

1. изучить литературу о многочленах;
2. применить теорию высшей алгебры в решении задач элементарной математики;
3. составить программы для нахождения частного и остатка при делении многочленов, наибольшего общего делителя двух многочленов, производной многочлена; разложения многочленов на кратные множители.

# 1. Многочлены

Общий вид уравнения n-ной степени (где n некоторое положительное число) есть

. (1.1)

Коэффициенты  этого уравнения мы будем считать произвольными комплексными числами, причем старший коэффициент  должен быть отличным от нуля.

Если написано уравнение (1.1), то всегда предполагается, что требуется его решить, найти такие числовые значения для неизвестного *x*, которые удовлетворяют этому уравнению, то есть после подстановки вместо неизвестного и выполнения всех указанных операций обращают левую часть уравнения (1.1) в нуль.

Целесообразно заменить задачу решения уравнения (1.1) более общей задачей изучения левой части этого уравнения

, (1.2)

называемой *многочленом n-ной степени от неизвестного х*. Многочленом называется лишь выражение вида (1.2), то есть лишь сумма целых неотрицательных степеней неизвестного *x*, взятых с некоторыми числовыми коэффициентами. В частности, мы не будем считать многочленами такие выражения, которые содержат неизвестное *x* с отрицательными или дробными показателями. Для сокращенной записи многочленов употребляются символы *f(x), g(x)* и так далее.

# 2. Деление многочленов

Теория многочленов в определенном отношении похожа на теорию целых чисел, хотя внешне эти две теории не имеют ничего общего. Внутренняя же близость, схожесть этих теорий объясняется тем, что для многочленов, так же как и для целых чисел, можно определить деление и, что еще более важно, деление с остатком.

##

## 2.1. Делимость многочленов. Свойства делимости

*Многочлен  делится на многочлен , если существует такой многочлен , что выполняется равенство*

** (2.1)

Например, из равенства ** следует, что делится на многочлен и на многочлен .

Многочлен  в равенстве (2.1) определяется однозначно. Если бы существовал многочлен , удовлетворяющий равенству (2.1), то мы получили бы, что

**  (2.2)

откуда 

Но многочлен  по условию ненулевой, и в силу утверждения или  нулевом является многочлен , т.е. многочлен совпадает с .

Многочлен  в равенстве (2.1) называется *частным* от деления  на , а  – делителем.

Укажем некоторые основные свойства делимости многочленов.

**1**. Если  делится , а  делится на , то  будет делиться на .

В самом деле, по условию  и , а поэтому .

**2**. Если  и  делятся на , то их сумма и разность также делятся на .

Из равенств и  вытекает .

**3**.Если  делится на , то произведение  на любой многочлен  также будет делиться на .

Если , то .

Из 2. и 3. вытекает следующее свойство:

**4**. Если каждый из многочленов  делится на , то на  будет делиться и многочлен , где  - произвольные многочлены.

**5**. Всякий многочлен  делится на любой многочлен нулевой степени.

Если , а *с* - произвольное число, не равное нулю, то есть произвольный многочлен нулевой степени, то .

**6**. Если  делится на , то  делится и на *с*, где *с* – произвольное число отличное от нуля.

Из равенства  следует равенство .

**7**. Многочлены , , и только они будут делителями многочлена , имеющими такую же степень, что и .

Действительно, . То есть  делится на .

Если  делится на , причем степени  и  совпадают, то степень частного от деления  на  должна быть равной нулю, то есть , , откуда .

Отсюда вытекает следующее свойство:

**8**. Тогда и только тогда многочлены ,  одновременно делятся друг на друга, если , .

Из 1. и 8. вытекает свойство:

**9**. Всякий делитель одного из двух многочленов , , где , будет делителем и для другого многочлена.

Свойства делимости многочленов могут быть применены для изучения делимости в множестве целых чисел. Выясним, например, *для каких целых чисел n числоявляется простым.*

Натуральное число, отличное от 1, называется простым, если оно делится только на 1 и на само себя; целое отрицательное число k называется простым, если число –k простое.

Для ответа на поставленный вопрос заметим, что справедливо равенство

 (2.3)

и поэтому число делится на  и на  Следовательно, оно может быть простым только в случае, когда один из этих делителей равен 1 или –1, т.е. выполняется хотя бы одно из равенств   

Остается проверить следующие значения *n*: 3, 1, 0, -3, -1 и –2. При этих значениях *n* рассматриваемое число равно соответственно 19, -5, 3, 4, так что искомое множество чисел есть 

Может возникнуть вопрос: откуда взялось равенство (2.3)? Как мы догадались, что многочлен  таким образом раскладывается на множители? Для нахождения разложений такого типа необязательно прибегать к искусственным группировкам, это можно сделать с помощью теории, которая будет изложена ниже.

Из этого примера видно, что уже для решения задач, связанных с делимостью целых чисел, полезно уметь выяснять, делится ли данный многочлен на некоторый другой многочлен (раскладывается ли на множители).Ответ на такой и многие другие вопросы можно найти с помощью деления многочлена с остатком.

##

## 2.2. Деление многочленов с остатком

Для многочленов, как и для целых чисел, существует алгоритм деления с остатком.

Теорема о делении с остатком. *Для любых двух многочленов f(x) и g(x) можно найти такие многочлены q(x) и r(x , что*

*f(x)=g(x)q(x)+r(x),*

*причем степень r(x) меньше степени g(x) или же r(x)=0. Многочлены q(x) и r(x), удовлетворяющие этому условию, определяются однозначно.*

Если разности *f(x)-r(x) и * обе делятся на *g(x),* то их разность ** также делится на *g(x).* Если бы многочлен *s(x)* был ненулевым, то он имел бы степень меньшую, чем  *g(x),* и не мог бы тогда делится на *g(x).* Следовательно, *s(x)=0,* так что .

В практической деятельности для нахождения частного и остатка применяют способ вычисления, называемый «деление углом». Покажем его на примере.

**Пример.** Найти частный и остаток от деления  на .

**1.**  и 

 |

 











Частным от деления  на  является многочлен , остатком – .

**2.**  и 

 │

 







Частным от деления  на  является многочлен , остатком – .

Это правило в общем виде можно сформулировать так:

1) разделить старший член многочлена *f(x)* на старший член *g(x)* и записать результат «под длинной стороной угла»;

2)умножить *g(x)* на результат действия 1) и записать произведение под многочленом *f(x)*;

3) вычесть из *f(x)* записанный под ним многочлен;

4) проверить имеет ли результат действия 3) степень меньшую, чем степень *g(x)*; если да (или результат нулевой), то он является остатком, а под длинной стороной угла записано частное, если нет, то применить к этому результату действие 1), рассматривая его как многочлен *f(x)*.

Я составила программу для нахождения частного и остатка.

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

 Dialogs, StdCtrls, Grids;

type

 TForm1 = class(TForm)

 SGd1: TStringGrid;

 Edit1: TEdit;

 Edit2: TEdit;

 Edit3: TEdit;

 Edit4: TEdit;

 Edit5: TEdit;

 Edit6: TEdit;

 Button1: TButton;

 Label1: TLabel;

 Label2: TLabel;

 Label3: TLabel;

 Label4: TLabel;

 Label5: TLabel;

 Label6: TLabel;

 Label7: TLabel;

 Label8: TLabel;

 Label9: TLabel;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

 i,n,m,step,j,f:integer;

 d,l,s:string;

 a,a2,b,b2,k:array[0..100] of integer;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

 n:=StrToInt(Edit1.Text);

 m:=StrToInt(Edit2.Text);

for i:=n+1 downto 1 do begin

 a[i]:=StrToInt(SGd1.Cells[n-(i-1),0]);

end;

for i:=m+1 downto 1 do begin

 b[i]:=StrToInt(SGd1.Cells[m-(i-1),1]);

end;

s:='f1(x)=';

for i:=n+1 downto 1 do begin

 if a[i]<>0 then begin if(a[i-1]<0)or(i=1) then begin

 str(a[i],l);

 str(i-1,d);

 s:=s+l+'x^'+d;

 end

 else begin

 str(a[i],l);

 str(i-1,d);

 s:=s+l+'x^'+d+'+';

 end;

end;

end;

Edit3.Text:=s;

s:='f2(x)=';

for i:=m+1 downto 1 do begin

 if b[i]<>0 then begin if(b[i-1]<0)or(i=1) then begin

 str(b[i],l);

 str(i-1,d);

 s:=s+l+'x^'+d;

 end

 else begin

 str(b[i],l);

 str(i-1,d);

 s:=s+l+'x^'+d+'+';

 end;

end;

end;

Edit4.Text:=s;

for j:=N+1 downto 1 do begin

 a2[j]:=a[j];

 b2[j]:=0;

end;

step:=n-m;

f:=n+2;

for i:=step+1 downto 1do begin

 f:=f-1;

 k[i]:=a2[f];

for j:=m+1 downto 1do begin

 b2[j]:=k[i]\*b[j];

end;

for j:=f downto 1 do begin

 a2[j]:=a2[j]\*b[m+1];

end;

for j:=f downto 1 do begin

 a2[j]:=a2[j]-b2[j+1-i];

 b2[j]:=0;

end;

end;

s:='h(x)=';

for i:=f downto 1 do begin

 if k[i]<>0 then begin if(k[i-1]<0)or(i=1) then begin

 str(k[i],l);

 str(i-1,d);

 s:=s+l+'x^'+d;

 end

 else begin

 str(k[i],l);

 str(i-1,d);

 s:=s+l+'x^'+d+'+';

 end;

end;

end;

Edit5.Text:=s;

s:='r(x)=';

for i:=n downto 0 do begin

 if a2[i]<>0 then begin if(a2[i-1]<0)or(i=1) then begin

 str(a2[i],l);

 str(i-1,d);

 s:=s+l+'x^'+d;

 end

 else begin

 str(a2[i],l);

 str(i-1,d);

 s:=s+l+'x^'+d+'+';

 end;

end;

end;

Edit6.Text:=s;

end;

end.


##

## 2.3. Наибольший общий делитель многочленов

Пусть даны произвольные многочлены  и . Многочлен будет называться *общим делителем* для  и , если он служит делителем для каждого из этих многочленов. Свойство 5. показывает, что к числу общих делителей многочленов  и  принадлежат все многочлены нулевой степени. Если других общих делителей эти два многочлена не имеют, то они называются *взаимно простыми*.

В общем же случае многочлены  и  могут обладать делителями, зависящими от , и введем понятие о *наибольшем* общем делителе этих многочленов.

*Наибольшим общим делителем* отличных от нуля многочленов  и  называется такой многочлен , который является их общим делителем и, вместе с тем, сам делится на любой другой общий делитель этих многочленов. Обозначается наибольший общий делитель многочленов  и  символом .

Это определение оставляет открытым вопрос, существует ли наибольший общий делитель для любых многочленов  и . Ответ на этот вопрос положительный. Существует метод для практического разыскания наибольшего общего делителя данных многочленов, называемый *алгоритмом последовательного деления* или *алгоритмом Евклида.*

##

## 2.4. Алгоритм Евклида

**Алгоритм Евклида** **–** метод для нахождения наибольшего общего делителя двух целых чисел, а также двух многочленов от одного переменного. Он первоначально был изложен в «Началах» Евклида в геометрической форме как способ нахождения общей меры двух отрезков. Алгоритм Евклида для нахождения наибольшего общего делителя, как в кольце целых чисел, так и в кольце многочленов от одного переменного является частным случаем некого общего алгоритма в евклидовых кольцах.

Алгоритм Евклида для нахождения наибольшего общего делителя двух многочленов  и  состоит в последовательном делении с остатком  на *,* затем  на первый остаток **,затем ** на второй остаток и так далее. Так как степени остатков все время понижаются, то в этой цепочке последовательных делений мы дойдем до такого места, на котором деление совершится нацело и процесс остановится. Последний отличный от нуля остаток , на который нацело делится предыдущий остаток , и является наибольшим общим делителем многочленов  и *.*

Для доказательства запишем изложенное в виде следующей цепочки равенств:



(2.4)



Последнее равенство показывает, что  служит делителем для . Отсюда следует, что оба слагаемых правой части предпоследнего равенства делятся на , а поэтому  будет делителем и для . Далее, таким же путем, поднимаясь вверх, мы получим, что  является делителем и для , …, , . Отсюда ввиду второго равенства, будет следовать, что  служит делителем для , а поэтому, на основании первого равенства, - и для .

Возьмем теперь произвольный общий делитель  многочленов  и . Так как левая часть и первое слагаемое правой части первого из равенств делятся на , то  также будет делится на . Переходя ко второму и следующему равенствам, таким же способом получим, что на  делятся многочлены , , … Наконец, если уже будет доказано, что  и  делятся на , то из предпоследнего равенства получим, что  делится на . Таким образом,  на самом деле будет наибольшим общим делителем для  и .

Мы доказали, что любые два многочлена обладают наибольшим общим делителем, и получили способ его вычисления. Этот способ показывает, что *если многочлены*   и  *имеют оба рациональные или действительные коэффициенты, то и коэффициенты их наибольшего общего делителя также будут рациональными или, соответственно, действительными,* хотя у этих многочленов могут существовать и такие делители, не все коэффициенты которых рациональны (действительны).

Если  есть наибольший общий делитель многочленов  и , то, как показывают свойства 8. и 9., в качестве наибольшего общего делителя этих многочленов можно было бы выбрать также многочлен , где  - произвольное число, отличное от нуля. Иными словами, *их наибольший общий делитель двух многочленов определен лишь с точностью до множителя нулевой степени*. Ввиду этого можно условиться, что старший коэффициент наибольшего общего делителя двух многочленов будет всегда считаться равным единице. Используя это условие можно сказать, что *два многочлена тогда и только тогда взаимно просты, если их наибольший общий делитель равен единице*. В самом деле, в качестве наибольшего общего делителя двух взаимно простых многочленов можно взять любое число, отличное от нуля, но, умножая его на обратный элемент, получим единицу.

Применяя алгоритм Евклида к многочленам с целыми коэффициентами, можем, чтобы избежать дробных коэффициентов, умножить делимое или сократить делитель на любое не равное нулю число, причем не только начиная какое-либо из последовательных делений, но и в процессе самого этого деления. Это будет приводить к искажению частного, но интересующие нас остатки будут приобретать лишь некоторый множитель нулевой степени, что при разыскании наибольшего общего делителя допускается.

**Пример.** Найти наибольший общий делитель многочленов  и .

**1.**  и 

Совершим требуемые деления с остатком:

 | 

 













 | 

 













 | 

 







Построение последовательности Евклида закончено. Ее последний член  есть наибольший общий делитель исходных многочленов.

**2.**  и 

Совершим требуемые деления с остатком:

 │

 ‌‌‌‌‌‌‌‌‌ || ‌







 │

 













 │

 







Построение последовательности Евклида закончено. Ее последний член  есть наибольший общий делитель исходных многочленов.

Я составила программу для нахождения наибольшего общего делителя двух многочленов:

unit Unit1;

interface

uses

 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

 Dialogs, StdCtrls, Grids;

type

 TForm1 = class(TForm)

 Label1: TLabel;

 Label2: TLabel;

 Edit1: TEdit;

 Edit2: TEdit;

 SGd1: TStringGrid;

 Label3: TLabel;

 Button1: TButton;

 Label4: TLabel;

 Edit4: TEdit;

 Label5: TLabel;

 Label6: TLabel;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

 st1,st2:integer;

 kof1,kof2,k1,k2:array[0..10] of integer;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var i,j,k\_1,st3,l:integer;

 sokr:boolean;

 k2\_2,k1\_1:array[0..10] of integer;

begin

 st1:=StrToInt(Edit1.Text);

 st2:=StrToInt(Edit2.Text);

 for i:=0 to st1 do begin

 kof1[i]:=StrToInt(SGd1.Cells[i,0]);

 k1[i]:=StrToInt(SGd1.Cells[i,0]);

 end;

 for i:=0 to st2 do begin

 kof2[i]:=StrToInt(SGd1.Cells[i,1]);

 k2[i]:=StrToInt(SGd1.Cells[i,1]);

 end;

 while kof2[0]<>0 do begin

 repeat

 Edit4.Text:='';

 k\_1:=k1[0];

 if k1[0]<>kof2[0] then begin

 if (k1[0] mod kof2[0])=0 then begin

 for j:=0 to st2 do

 k2[j]:=(k1[0] div kof2[0])\*kof2[j];

 end

 else begin

 if k2[0]<>1 then

 for j:=0 to st1 do

 k1[j]:=kof2[0]\*k1[j];

 if k\_1<>1 then begin

 for j:=0 to st2 do

 k2[j]:=k\_1\*kof2[j];

 end;

 end;

 end;

 for i:=1 to st1 do begin

 k1[i-1]:=k1[i]-k2[i];

 end;

 st1:=st1-1;

 until st1<st2;

 if k1[0]<>0 then begin //Сокращение

 sokr:=true;

 for i:=1 to st1 do

 if k1[i]<>0 then begin

 if (k1[i] mod k1[0])<>0 then sokr:=false;

 end;

 k\_1:=k1[0];

 if sokr=true then

 for i:=0 to st1 do

 k1[i]:=k1[i] div k\_1;

 end;

 for i:=0 to st2 do //Замена многочленов

 k2\_2[i]:=kof2[i];

 for i:=0 to st1 do

 k1\_1[i]:=k1[i];

 for i:=0 to 10 do begin

 kof1[i]:=0;

 k1[i]:=0;

 kof2[i]:=0;

 k2[i]:=0;

 end;

 for i:=0 to st2 do begin

 k1[i]:=k2\_2[i];

 if k1[i]<>0 then

 Edit4.Text:=Edit4.Text+IntToStr(k1[i])+'x^'+IntToStr(st2-i);

 if (k2\_2[i+1]>0)and(i<st2) then Edit4.Text:=Edit4.Text+'+';

 end;

 for i:=0 to st1 do begin

 k2[i]:=k1\_1[i];

 kof2[i]:=k1\_1[i];

 end;

 st3:=st1;

 st1:=st2;

 st2:=st3;

 end;

end;

end.

Используем алгоритм Евклида для доказательства следующей теоремы:

**Теорема.** Если  есть наибольший общий делитель многочленов  и , то можно найти такие многочлены  и , что

. (2.5)

Можно считать при этом, если степени многочленов  и  больше нуля, что степень  меньше степени , а степень  меньше степени .

*Доказательство* основано на равенствах (2.4). Если учтем, что , и положим , , то предпоследнее из равенств (2.4) даст:



.

Подставляя сюда выражение  через  и  из предшествующего равенства (2.4), получим:

,

где , . Продолжая подниматься вверх по равенствам (2.4), придем к доказываемому равенству (2.5).

Для доказательства второго утверждения теоремы предположим, что многочлены  и , удовлетворяющие равенству (2.5), уже найдены, но, например, степень  больше или равна степени . Делим  на :

,

где степень  меньше степени , и подставляем это выражение в (2). Получим равенство:



.

Степень множителя, стоящего при , уже меньше степени . Степень многочлена, стоящего в квадратных скобках, будет в свою очередь меньше степени , так как в противном случае степень второго слагаемого левой части была бы не меньше степени , а так как степень первого слагаемого меньше степени этого произведения, то вся левая часть имела бы степень, большую или равную степени , тогда как многочлен  заведомо имеет, при наших предположениях, меньшую степень.

Теорема доказана.

Одновременно получаем, что если многочлены  и  имеют рациональные или действительные коэффициенты, то и многочлены  и , удовлетворяющие равенству (2.5), можно подобрать так, что их коэффициенты будут рациональными или, соответственно действительными.

Применяя доказанную теорему к взаимно простым многочленам, получаем такой результат:

*Многочлены*  и  *тогда и только тогда взаимно просты, если можно найти многочлены  и , удовлетворяющие равенству*

. (2.6)

Опираясь на этот результат, можно доказать несколько простых, но важных теорем о взаимно простых многочленах:

**Теорема 1.**  Если многочлен  взаимно прост с каждым из многочленов  и , то он взаимно прост и с их произведением.

*Доказательство.* В самом деле, существуют, по (2.6), такие многочлены  и , что .

Умножая это равенство на , получаем:

,

откуда следует, что всякий общий делитель  и  был бы делителем и для ; однако по условию .

**Теорема 2.** Если произведениемногочленов  и  делится на , но  и  взаимно просты, то  делится на .

*Доказательство.* Умножая равенство  на , получим: .

Оба слагаемых левой части этого равенства делятся на ; на него делится, следовательно, и .

**Теорема 3.** Если многочлен  делится на каждый из многочленов  и , которые между собой взаимно просты, то  делится и на их произведение.

*Доказательство.* Действительно, , так что произведение, стоящее справа, делится на . Поэтому, по теореме 2,  делится на , , откуда .

Определение наибольшего общего делителя может быть распространен на случай любой конечной системы многочленов: *наибольшим общим делителем многочленов * называется такой общий делитель этих многочленов, который делится на любой другой общий делитель этих многочленов. Существование наибольшего общего делителя для любой конечной системы многочленов вытекает из следующей теоремы, дающей также способ его вычисления.

**Теорема.** Наибольший общий делитель многочленов  равен наибольшему общему делителю многочлена  и наибольшего общего делителя многочленов .

*Доказательство.* В самом деле, при  теорема очевидна. Примем поэтому, что для случая  она справедлива, то есть, в частности, уже доказано существование наибольшего общего делителя  многочленов . Обозначим через  наибольший общий делитель многочленов  и . Он будет общим делителем для всех заданных многочленов. С другой стороны, всякий другой общий делитель этих многочленов будет делителем также и для , а поэтому и для .

В частности, система многочленов  называется *взаимно простой*, если общими делителями этих многочленов являются лишь многочлены нулевой степени, то есть если их наибольший общий делитель равен 1. Если , то попарно эти многочлены могут и не быть взаимно простыми.

# 3. Кратные корни

**Теорема Безу**. *Многочлен f(x) делится на x-c тогда и только тогда, когда число c является его корнем.*

Рассмотрим произвольный многочлен *f(x)* и разделим его с остатком на двучлен *x-c*. Поскольку степень этого двучлена равна 1, то остаток либо равен 0, либо имеет степень 0. И в том, и в другом случае остаток *r* есть число. Таким образом, многочлен *f(x)* представляется в виде:

*f(x)= (x-c) q(x)+ r.*

Положив в этом тождестве *x= c*, получим что *f(c)=r.* Мы доказали тем самым, что остаток от деления многочлена на двучлен *x- c* равен значению многочлена при *x=c.*

С помощью теоремы Безу решим несколько задач.

**Пример 1**. Решить уравнение .

Многочлен *f(x)=*  имеет корень 2. По теореме Безу *f(x)* делится на *x-2*, то есть имеет место равенство

.

| 

 











Остается решить квадратное уравнение .

Это уравнение не имеет действительных корней, так что *x=2* – единственный действительный корень исходного уравнения.

**2**. Решить уравнение .

Многочлен *f(x)=*  имеет корень -2. По теореме Безу *f(x)* делится на *x+2*, то есть имеет место равенство .

 | 

 







0

Остается решить квадратное уравнение .

Это уравнение имеет корень 1. Так что *x=-2* и *x=1* – корни исходного уравнения.

Если *c* – корень многочлена *f(x)*, то есть *f(c)=0*, то *f(x)* делится на *x-c.* Может оказаться, что многочлен *f(x)* делится не только на первую степень линейного двучлена *x-c*, но и на более высокие его степени. Во всяком случае, найдется такое натуральное число *k*, что *f(x)* нацело делится на , но не делится на . Поэтому

,

где многочлен  на *x-c* уже не делится, то есть число *с* своим корнем не имеет. Число *k* называется *кратностью* корня *c* в многочлене *f(x)*, а сам корень *c* – *k- кратным корнем* этого многочлена. Если *k*=1, то говорят, что корень *с* –  *простой*.

# 4. Производная от многочлена

Понятие кратного корня тесно связано с понятием производной от многочлена. Мы изучаем многочлены с любыми комплексными коэффициентами и поэтому не можем просто воспользоваться понятием производной, введенным в курсе математического анализа. То, что будет сказано ниже, следует рассматривать как независимое от курса анализа определение производной многочлена.

Пусть дан многочлен n–ной степени

*f(x)=* 

с любыми комплексными коэффициентами. Его *производной (первой производной)* называется многочлен (n- 1)-й степени



Производная от многочлена нулевой степени и от нуля считается равной нулю. Производная от первой производной называется *второй производной* от многочлена *f(x)* и обозначается через *f“(x)* . Очевидно, что



и по этому , то есть (n+1)-я производная от многочлена n–й степени равна нулю.

Свойства, являющиеся формулами дифференцирования для суммы и произведения:

1. ** (4.1)

2. **  (4.2)

Эти формулы легко проверить, впрочем, непосредственным подсчетом, беря в качестве и два произвольных многочлена и применяя данное выше определение производной.

Формула (4.2) распространяется на случай произведения любого конечногочисла множителей, а поэтому выводится формула для производной от степени:

3. **  (4.3)

*Доказательство.* Используем метод математической индукции.

 

 

 

.

*Если число с является k –кратным корнем многочлена f(x), то при k>1 оно будет (k-1)–кратным корнем первой производной этого многочлена; если же k=1 , то с не будет служить корнем для .*

В самом деле, пусть

,  *,* (4.4)

где  уже не делится на *х-с*. Дифференцируя равенство (4.4), получаем:

.

Первое слагаемое суммы делится на *х-с*, а второе на *х-с* не делится; поэтому вся эта сумма на  *х-с* не может делиться. Учитывая, что частное от деления *f(x)* на ** определено однозначно, мы получаем, что ** является наибольшей степенью двучлена *х-с*, на которую делится многочлен .

Применяя эту теорему несколько раз, мы получаем, что *k-кратный корень многочлена f(x) будет (k-s)-кратным в s-й производной этого многочлена  и впервые не будет служить корнем для k-й производной от f(x).*

**Пример.** Найти производную  многочлена .



.

Я составила программу для нахождения первой производной многочлена.

unit Unit1;

interface

uses

 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

 Dialogs, StdCtrls, Grids;

type

 TForm1 = class(TForm)

 Edit1: TEdit;

 Label1: TLabel;

 SGd1: TStringGrid;

 Label2: TLabel;

 Button1: TButton;

 Edit2: TEdit;

 Edit3: TEdit;

 Label3: TLabel;

 Label4: TLabel;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

 c,i,st:integer;

 k,l,s:string;

 kof:array[0..100] of integer;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

 st:=StrToInt(Edit1.Text);

 for i:=0 to st do begin

 if SGd1.Cells[i,0]<>'' then

 kof[st-i]:=StrToInt(SGd1.Cells[i,0])

 else MessageDlg ('Внимание! Не введены значения коэффициентов!',mtWarning,[mbOK],0);

 end;

 s:='f(x)=';

 for i:=st downto 0 do begin

 if kof[i]<>0 then begin

 if(kof[i-1]<0)or(i=0) then begin

 str(kof[i],l);

 str(i,k);

 s:=s+l+'x^'+k;

 end

 else begin

 str(kof[i],l);

 str(i,k);

 s:=s+l+'x^'+k+'+';

 end;

 end;

 kof[i]:=kof[i]\*i;

 end;

 Edit2.Text:=s;

 s:='f1(x)=';

 for i:=st downto 0 do begin

 if kof[i]<>0 then begin

 if(kof[i-1]<0)or(i=1) then begin

 str(kof[i],l);

 str(i-1,k);

 s:=s+l+'x^'+k;

 end

 else begin

 str(kof[i],l);

 str(i-1,k);

 s:=s+l+'x^'+k+'+';

 end;

 end;

 Edit3.Text:=s;

 end;

end;

end.

**5. Кратные множители**

Существуют методы, позволяющие узнать, обладает ли данный многочлен кратными множителями, и в случае положительного ответа дающие возможность свести изучение этого многочлена к изучению многочленов, уже не содержащих кратных множителей.

**Теорема**. Если  является  - кратным неприводимым множителем многочлена , , то он будет - кратным множителем производной этого многочлена. В частности, простой множитель многочлена. Не входит в разложение производной.

В самом деле, пусть

, (5.1)

причем  уже не делится на . Дифференцируя равенство (5.1), получаем:

.

Второе из слагаемых, стоящих в скобках, не делится на . Действительно,  не делится по условию,  имеет меньшую степень, т.е. также не делится на . С другой стороны, первое слагаемое суммы, стоящей в квадратных скобках, делиться на, т.е. множитель , на самом деле входит в  с кратностью .

Из данной теоремы и из указанного выше способа разыскания наибольшего общего делителя двух многочленов следует, что если дано разложение многочлена  на неприводимые множители:

, (5.2)

то наибольший общий делитель многочлена  и его производной обладает следующим разложением на неприводимые множители:

, (5.3)

где множитель  следует при  заменять единицей. В частности, многочлен  тогда и только тогда не содержит кратных множителей, если он взаимно прост со своей производной.

##

## 5.1. Выделение кратных множителей

Если дан многочлен  с разложением (5.2) и если через  мы обозначим наибольший общий делитель  и его производной  то (5.3) будет разложением для . Деля (5.2) на (5.3), мы получим:



т.е. получим многочлен, не содержащий кратных множителей, причем всякий неприводимый множитель для , имеющего вообще говоря, меньшую степень и, во всяком случае, содержащего лишь простые множители. Если эта задача для  будет решена, то останется определить лишь кратность найденных неприводимых множителей в , что достигается применением алгоритма деления.

Усложняя изложенный сейчас метод, можно сразу перейти к рассмотрению нескольких многочленов без кратных множителей, причем, найдя неприводимые множители этих многочленов, мы не только найдем все неприводимые множители для , но и будем знать их кратность.

Пусть (5.2) будет разложением  на неприводимые множители, причем наивысшая кратность множителей есть , . Обозначим через  произведение всех однократных множителей многочлена , через  - произведение всех двукратных множителей, но взятых лишь по одному разу, и т.д., наконец  - произведение всех -кратных множителей, также взятых лишь по одному разу; если при этом для некоторого  в  отсутствуют -кратные множители, то полагаем . Тогда  будет делиться на - тую степень многочлена  и разложение (5.2) примет вид



а разложение (5.3) для  перепишется в виде 

обозначая через  наибольший общий делитель многочлена  и его производной и вообще через  наибольший общий делитель многочленов  и , таким путем получим:





……………………………



.

Отсюда

,





……………………………

,

И поэтому, наконец,

, , …, 

Таким образом, пользуясь лишь приемами, не требующими знания неприводимых множителей многочлена , а именно взятием производной, алгоритмом Евклида и алгоритмом деления, мы можем найти многочлены  без кратных множителей, причем всякий неприводимый множитель многочлена  , будет -кратным для .

**Пример.** Разложить многочлен  на кратные множители.







 │ 

 





 │ 

 













 │ 

 















 │

 













 │

 

















│

 

















│

 



















│

 











│

 















Многочлен  имеет разложение в виде .

Я составила программу для разложения многочлена на кратные множители.

unit Unit1;

interface

uses

 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

 Dialogs, StdCtrls, Grids;

type

 TForm1 = class(TForm)

 Edit1: TEdit;

 Label1: TLabel;

 SGd1: TStringGrid;

 Label2: TLabel;

 Button1: TButton;

 Label3: TLabel;

 SGd2: TStringGrid;

 SGd3: TStringGrid;

 SGd4: TStringGrid;

 Edit6: TEdit;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

 c,i,st1,st2,stiz,n\_iz,n\_nod,n,m,d\_st,step,f:integer;

 k,d,s:string;

 kof1,kof2,k1,k2,izubst,a,b,a2,b2,buf,est,fxst:array[0..15] of integer;

 izub,e,fx:array[0..50,0..50] of integer;

 first:boolean;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var i,j,k\_1,st3,l:integer;

 sokr:boolean;

 k2\_2,k1\_1:array[0..10] of integer;

begin

 st1:=StrToInt(Edit1.Text);

 for i:=0 to st1 do begin

 SGd4.Cells[i,0]:=SGd1.Cells[i,0];

 end;

 repeat

 n\_iz:=n\_iz+1;

 st2:=st1-1;

 for i:=0 to st1 do begin

 if SGd1.Cells[i,0]<>'' then

 kof1[st1-i]:=StrToInt(SGd1.Cells[i,0])

 else MessageDlg('Внимание! Не введены значения коэффициентов!',mtWarning,[mbOK],0);

 end;

 s:='f(x)=';

 for i:=st1 downto 0 do begin

 if kof1[i]<>0 then begin

 if(kof1[i-1]<0)or(i=0) then begin

 str(kof1[i],d);

 str(i,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(kof1[i],d);

 str(i,k);

 s:=s+d+'x^'+k+'+';

 end;

 end;

 kof2[i-1]:=kof1[i]\*i;

 end;

 //Edit2.Text:=s;

 s:='f1(x)=';

 for i:=st2 downto 0 do begin

 SGd2.Cells[st2-i,0]:=inttostr(kof2[i]);

 if kof2[i]<>0 then begin

 if(kof2[i-1]<0)or(i=1) then begin

 str(kof2[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(kof2[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 end;

 //Edit3.Text:=s;

 end;

 for i:=0 to st1 do begin

 kof1[i]:=StrToInt(SGd1.Cells[i,0]);

 k1[i]:=StrToInt(SGd1.Cells[i,0]);

 end;

 for i:=0 to st2 do begin

 kof2[i]:=StrToInt(SGd2.Cells[i,0]);

 k2[i]:=StrToInt(SGd2.Cells[i,0]);

 end;

 while kof2[0]<>0 do begin

 repeat

 //Edit4.Text:='';

 stiz:=0;

 k\_1:=k1[0];

 if k1[0]<>kof2[0] then begin

 if (k1[0] mod kof2[0])=0 then begin

 for j:=0 to st2 do

 k2[j]:=(k1[0] div kof2[0])\*kof2[j];

 end

 else begin

 if k2[0]<>1 then

 for j:=0 to st1 do

 k1[j]:=kof2[0]\*k1[j];

 if k\_1<>1 then begin

 for j:=0 to st2 do

 k2[j]:=k\_1\*kof2[j];

 end;

 end;

 end;

 for i:=1 to st1 do begin

 k1[i-1]:=k1[i]-k2[i];

 end;

 st1:=st1-1;

 until st1<st2;

 if k1[0]<>0 then begin //Сокращение

 sokr:=true;

 for i:=1 to st1 do

 if k1[i]<>0 then begin

 if (k1[i] mod k1[0])<>0 then sokr:=false;

 end;

 k\_1:=k1[0];

 if sokr=true then

 for i:=0 to st1 do

 k1[i]:=k1[i] div k\_1;

 end;

 for i:=0 to st2 do //Замена многочленов

 k2\_2[i]:=kof2[i];

 for i:=0 to st1 do

 k1\_1[i]:=k1[i];

 for i:=0 to 10 do begin

 SGd3.Cells[i,0]:='';

 SGd1.Cells[i,0]:='';

 kof1[i]:=0;

 k1[i]:=0;

 kof2[i]:=0;

 k2[i]:=0;

 izub[n\_iz,i]:=0;

 end;

 izubst[n\_iz]:=st2;

 for i:=0 to st2 do begin

 k1[i]:=k2\_2[i];

 SGd1.Cells[i,0]:=inttostr(k1[i]);

 izub[n\_iz,i]:=k1[i];

 if k1[i]<>0 then begin

 //Edit4.Text:=Edit4.Text+IntToStr(k1[i])+'x^'+IntToStr(st2-i);

 end;

 if (k2\_2[i+1]>0)and(i<st2) then //Edit4.Text:=Edit4.Text+'+';

 end;

 for i:=0 to st1 do begin

 k2[i]:=k1\_1[i];

 kof2[i]:=k1\_1[i];

 end;

 st3:=st1;

 st1:=st2;

 st2:=st3;

 end;

 until (st1=0);

 d\_st:=StrToInt(Edit1.Text);

 for i:=d\_st+1 downto 1 do begin

 kof1[i]:=StrToInt(SGd4.Cells[d\_st-(i-1),0]);

 end;

 //Нахождение Е

 first:=true;

 for n\_nod:=1 to n\_iz do begin

 n:=d\_st;

 m:=izubst[n\_nod];

 d\_st:=m;

 for i:=n+1 downto 1 do begin

 a[i]:=kof1[i];

 end;

 for i:=m+1 downto 1 do begin

 b[i]:=izub[n\_nod,m-(i-1)];

 kof1[i]:=b[i];

 end;

 s:='f1(x)=';

 for i:=n+1 downto 1 do begin

 if a[i]<>0 then begin

 if(a[i-1]<0)or(i=1) then begin

 str(a[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(a[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 end;

 end;

 //Edit3.Text:=s;

 s:='f2(x)=';

 for i:=m+1 downto 1 do begin

 if b[i]<>0 then begin

 if(b[i-1]<0)or(i=1) then begin

 str(b[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(b[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 end;

 end;

 //Edit4.Text:=s;

 for j:=n+1 downto 1 do begin

 a2[j]:=a[j];

 b2[j]:=0;

 end;

 step:=n-m;

 f:=n+2;

 for i:=step+1 downto 1 do begin

 f:=f-1;

 buf[i]:=a2[f];

 for j:=m+1 downto 1 do begin

 b2[j]:=buf[i]\*b[j];

 end;

 for j:=f downto 1 do begin

 a2[j]:=a2[j]\*b[m+1];

 end;

 for j:=f downto 1 do begin

 a2[j]:=a2[j]-b2[j+1-i];

 b2[j]:=0;

 end;

 end;

 s:='h(x)=';

 for i:=f+1 downto 1 do begin

 e[n\_nod,i]:=buf[i];

 if buf[i]<>0 then begin

 if(buf[i-1]<0)or(i=1) then begin

 str(buf[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(buf[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 buf[i]:=0;

 end;

 end;

 est[n\_nod]:=f;

 //Edit5.Text:=s;

 s:='r(x)=';

 for i:=n downto 0 do begin

 if a2[i]<>0 then begin

 if(a2[i-1]<0)or(i=1) then begin

 str(a2[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(a2[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 end;

 end;

 Edit6.Text:=s;

 first:=false;

 end;

 for n\_nod:=1 to n\_iz-1 do begin

 n:=est[n\_nod];

 m:=est[n\_nod+1];

 d\_st:=m;

 for i:=n+1 downto 1 do begin

 a[i]:=e[n\_nod,i];

 end;

 for i:=m+1 downto 1 do begin

 b[i]:=e[n\_nod+1,i];

 kof1[i]:=b[i];

 if n\_nod=n\_iz-1 then fx[n\_iz,i]:=b[i];

 end;

 s:='f1(x)=';

 for i:=n+1 downto 1 do begin

 if a[i]<>0 then begin if(a[i-1]<0)or(i=1) then begin

 str(a[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(a[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 end;

 end;

 //Edit3.Text:=s;

 s:='f2(x)=';

 for i:=m+1 downto 1 do begin

 if b[i]<>0 then begin if(b[i-1]<0)or(i=1) then begin

 str(b[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(b[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 end;

 end;

 //Edit4.Text:=s;

 for j:=n+1 downto 1 do begin

 a2[j]:=a[j];

 b2[j]:=0;

 end;

 step:=n-m;

 f:=n+2;

 for i:=step+1 downto 1 do begin

 f:=f-1;

 buf[i]:=a2[f];

 for j:=m+1 downto 1 do begin

 b2[j]:=buf[i]\*b[j];

 end;

 for j:=f downto 1 do begin

 a2[j]:=a2[j]\*b[m+1];

 end;

 for j:=f downto 1 do begin

 a2[j]:=a2[j]-b2[j+1-i];

 b2[j]:=0;

 end;

 end;

 s:='h(x)=';

 for i:=f+1 downto 1 do begin

 fx[n\_nod,i]:=buf[i];

 if buf[i]<>0 then begin if(buf[i-1]<0)or(i=1) then begin

 str(buf[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(buf[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 buf[i]:=0;

 end;

 end;

 //Edit5.Text:=s;

 fxst[n\_nod]:=f;

 s:='r(x)=';

 for i:=n downto 0 do begin

 if a2[i]<>0 then begin if(a2[i-1]<0)or(i=1) then begin

 str(a2[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(a2[i],d);

 str(i-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 end;

 end;

 Edit6.Text:=s;

 end;

 fxst[n\_iz]:=est[n\_iz]+1;

 Edit6.Text:='';

 s:='';

 for i:=1 to n\_iz do begin

 s:=s+'(';

 for j:=fxst[i] downto 0 do begin

 if fx[i,j]<>0 then begin

 if(fx[i,j-1]<0)or(j=1) then begin

 str(fx[i,j],d);

 str(j-1,k);

 s:=s+d+'x^'+k;

 end

 else begin

 str(fx[i,j],d);

 str(j-1,k);

 s:=s+d+'x^'+k+'+';

 end;

 end;

 end;

 s:=s+')^'+IntToStr(i)+' ';

 Edit6.Text:=Edit6.Text+s;

 s:='';

 end;

 for i:=0 to 10 do begin

 SGd1.Cells[i,0]:=SGd4.Cells[i,0];

 end;

end;

end.


# Заключение

При выполнении дипломной работы я рассмотрела следующие вопросы:

– делимость многочленов;

– деление многочленов с остатком;

– наибольший общий делитель, алгоритм Евклида;

– кратные корни;

– кратные множители, выделение кратных множителей;

– производные от многочленов.

Составила программы для нахождения частного и остатка при делении многочленов; наибольшего общего делителя двух многочленов; производной многочлена.

# Список использованной литературы

1. Алгебра и теория чисел. Под ред. Н. Я. Виленкина. Москва: Просвещение, 1984.
2. Архангельский А. Я. Программирование в Delphi 6. Москва: ЗАО Бином, 2003.
3. Архангельский А. Я. Delphi 7. Справочное пособие. Москва: ООО Бином-Пресс, 2004.
4. Курош А. Г. Курс высшей алгебры. Москва: Наука, 1971.
5. Ляпин Е. С., Евсеев А. Е. Алгебра и теория чисел. Часть II. Линейная алгебра и полиномы. Москва: Просвещение, 1978.
6. Мантуров О. В. и др. Математика в понятиях, определениях и терминах. Часть 2. Москва: Просвещение, 1982.
7. Попов В.Б. Turbo Pascal. Москва: Финансы и статистика, 2000.
8. Потапов М. К., Александров В. В., Пасиченко П. И. Алгебра и анализ элементарных функций. Москва: Наука, 1980.
9. Сабинина Л. В. Математика в понятиях, определениях и терминах. Часть I. Москва: Просвещение, 1978.
10. Сборник задач по алгебре. Под ред. А. И. Кострикина. Москва: Наука, 1987.
11. Смолин Ю. Н. Алгебра и теория чисел. Перемь:1996.
12. Солодовников А. С., Родина М. А. Задачник-практикум по алгебре. Часть IV. Москва: Просвещение, 1985.
13. Фадеев Д. К. Лекции по алгебре. Москва: Наука, 1984.
14. Фадеев Д. К., Соминский И. С. Сборник задач по высшей алгебре. Москва: Наука, 1968.
15. Шварцбурд С. И. Избранные вопросы математики. Москва: Просвещение, 1980.