$$J = \sum_{\mu=n}^{m} \mu^{-\alpha-1} \left(\sum_{\nu=n}^{\mu} a_\nu v^\lambda \right)^p$$

белгілейік және $2^{N-1} < n \le 2^N$ және $2^M \le \mathrm{m} < 2^{M+1}$ теңсіздіктері орындалатындай M, N сандарын таңдайық. Онда

$$\begin{aligned} J &\ge \sum_{\mu=2^N}^{2^M} \mu^{-\alpha-1} \left(\sum_{\nu=n}^{\mu} a_\nu v^\lambda \right)^p \ge \\ &\ge \sum_{i=N}^{M-1} \sum_{\mu=2^i}^{2^{i+1}-1} \mu^{-\alpha-1} \left(\sum_{\nu=n}^{\mu} a_\nu v^\lambda \right)^p \ge \\ &\ge \sum_{i=N}^{M-1} \sum_{\mu=2^i}^{2^{i+1}-1} \mu^{-\alpha-1} \left(\sum_{\nu=n}^{2^i} a_\nu v^\lambda \right)^p = \\ &= \sum_{i=N}^{M-1} \left(\sum_{\nu=n}^{2^i} a_\nu v^\lambda \right)^p \sum_{\mu=2^i}^{2^{i+1}-1} \mu^{-\alpha-1} \ge \\ &\ge C_1 \sum_{i=N}^{M-1} \left(\sum_{\nu=n}^{2^i} a_\nu v^\lambda \right)^p 2^{-i\alpha}. \end{aligned}$$

Қолданылған әдебиеттер тізімі

1. Potapov M.K., Berisha F.M., Berisha N.Sh., Kadriu R. Some reverse lp-type ineualities involving quasi monotone sequences // Math. Inequal. Appl. – 2015. – V. 18, № 4. – P. 1245-1252.
2. Berisha F.M., Berisha N.Sh., Potapov M.K., Dema M. On Approximations by Trigonometric Polynomials of Classes of Functions Defined by Moduli of Smoothness // Hindawi Abst. Appl. Anal. – 2017. – V. 20, № 5. – P. 1–11.
3. Liflyand E., Tikhonov S., Zeltser M. Extending tests for convergence of number series // J. Math. Anal. Appl. – 2011. – V. 377, № 1. – P. 194-206.

УДК 519.61

УМНОЖЕНИЕ МНОГОЧЛЕНОВ В СРЕДЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

Аубакирова Д.А.

магистрант 2-курса

Евразийский национальный университет им. Л.Н Гумилева, г. Нур-Султан,

Республика Казахстан

E-mail: dana.aubakirova7@gmail.ru

Научный руководитель – к.ф.-м.н. Абуталипова Ш.У.

Аннотация:Цель статьи заключается в рассмотрении алгоритма умножения многочленов. Полученный в статье код, разработанный на базе программирования языка Python, несколько упрощает решение подобных задач. В статье также рассмотрены основные правила и понятия об умножении многочленов.

Abstract: The purpose of the article is to consider the polynomial multiplication algorithm. The code obtained in the article, developed on the basis of programming the Python language, somewhat simplifies the solution of such problems. The article also discusses the basic rules and concepts of multiplication of polynomials.

Ключевые слова: многочлен; умножение многочленов; переменные; Питон

Keywords: polynomial; polynomial multiplication; variables; Python

Как известно, самыми известными сайтами онлайн решения задач являются math-solution.ru, planetcalc.ru, и тому подобное. Сайты само собой требуют соответственных тех или иных ресурсов, так же они не могут отвечать всем запросам науки.

Возьмем две переменные x и y. Произведение a· xk· yl, где a – число, называется одночленом. Его степень равна k+l. Сумма одночленов называется многочленом. В отличие от многочленов с одной переменной, для многочленов с большим числом переменных нет общепринятой стандартной записи.

Так же, как и многочлены от одной переменной, многочлены от двух переменных могут раскладываться на множители.

Чтобы разработать код для алгоритма, необходимо сначала изучить и понять сам алгоритм.

Умножение полиномов

Зададим два многочлена a+b и c+d и выполним их умножение.

В первую очередь запишем произведение исходных многочленов: поставим между ними знак умножения, предварительно заключив многочлены в скобки. Получим: (a+b)· (c+d). Теперь обозначим множитель (c+d) как x, тогда выражение получит вид: (a+b)· x(a+b)·x, что по сути является произведением многочлена и одночлена. Осуществим умножение: (a+b)· x=a· x+b· x, а затем обратно заменим x на (c+d): a· (c+d)+b· (c+d). И вновь применив правило умножения многочлена на одночлен, преобразуем выражение в: a· c+a· d+b· c+b· d. Резюмируя: произведению заданных многочленов a+b и c+d соответствует равенство (a+b)· (c+d)=a· c+a· d+b· c+b· d.

Рассуждения, которые мы привели выше, дают возможность сделать важные выводы:

1. Результат умножения многочлена на многочлен - многочлен. Данное утверждение справедливо для любых перемножаемых многочленов.
2. Произведение многочленов есть сумма произведений каждого члена одного многочлена на каждый член другого. Откуда можно сделать заключение, что при умножении многочленов, содержащих m и n членов соответственно, указанная сумма произведений членов состоит из m· n слагаемых.

Теперь можем сформулировать правило умножения многочленов:

Для осуществления умножения многочлена на многочлен, необходимо каждый член одного многочлена умножить на каждый член другого многочлена и найти сумму полученных произведений.

Рассмотрим такой пример: (5x-2y) и (3x+8y).

Решением умножения этих двух многочленов будет: 15x2+34xy-16y2.

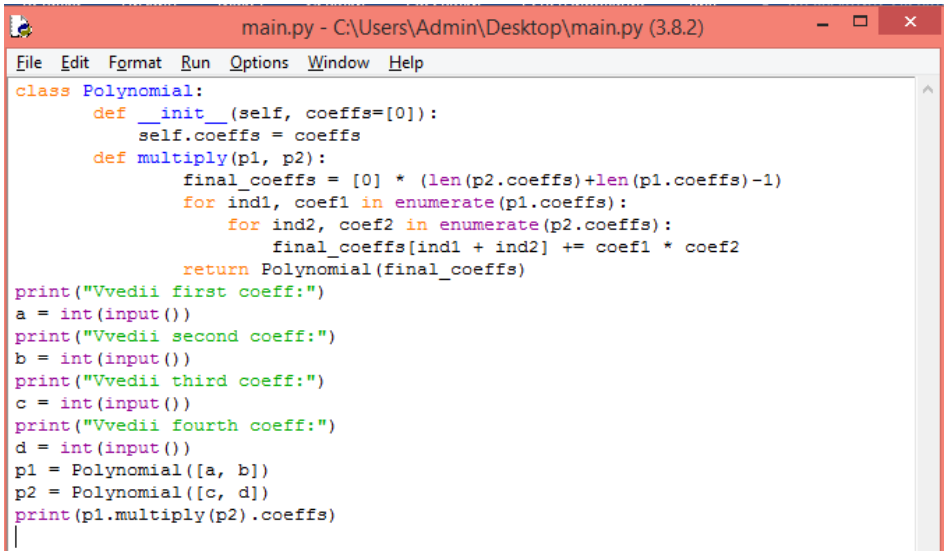
Если записать подробное решение по правилам:

(5x-2y) · (3x+8y) = 5x· 3x+5x· 8y-2y· 3x-2y· 8y =

= 15x2+40xy-6xy-16y2 = 15x2+34xy-16y2

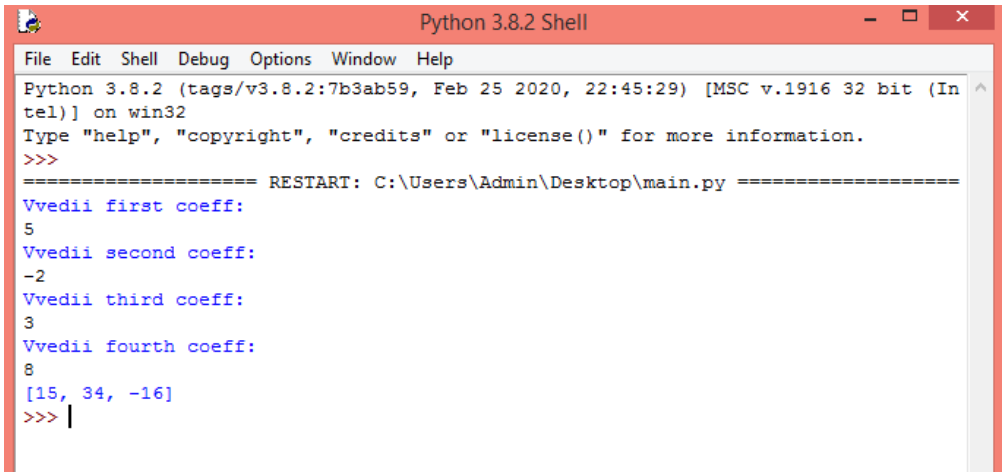
Запишем код в Python для решения этого уравнения, то есть для вычисления коэффициентов нового итогового многочлена полученного при результате умножения двух многочленов от двух переменных.

```
class Polynomial:
def __init__(self, coeffs=[0]):
self.coeffs = coeffs
def multiply(p1, p2):
final_coeffs = [0] * (len(p2.coeffs)+len(p1.coeffs)-1)
for ind1, coef1 in enumerate(p1.coeffs):
for ind2, coef2 in enumerate(p2.coeffs):
final_coeffs[ind1 + ind2] += coef1 * coef2
return Polynomial(final_coeffs)
print("Vvedii first coeff:")
a = int(input())
print("Vvedii second coeff:")
b = int(input())
print("Vvedii third coeff:")
c = int(input())
print("Vvedii fourth coeff:")
d = int(input())
p1 = Polynomial([a, b])
p2 = Polynomial([c, d])
print(p1.multiply(p2).coeffs)
```



Так он должен выглядеть при вводе: если вводить классы, операции неправильно, то исход будет безрезультатным.

Вводим коэффициенты заданных многочленов для умножения по порядку.



Как было отмечено, программа выдала нам итоговые коэффициенты: [15, 34, -16], сравним коэффициенты с ответом, полученным выше: 15x2+34xy-16y2. Как мы видим они абсолютно идентичны. Из этого следует заключить итог, что код абсолютно правильный и применим в соответствующих областях.

References

1. Python Crash Course, Eric Matthes.
2. Invent Your Own Computer Games with Python, Al Sweigart.
3. Математика на Python. Часть I. Элементы линейной алгебры и аналитической геометрии, И. А. Александрова, А. С. Балджы, М. Б. Хрипунова.
4. С.Т. Завало. Элементарная алгебра. Изд-во "Просвещение", М., 1964 г.

УДК 517

МАТРИЧНЫЕ ПРЕДСТАВЛЕНИЯ КОНЕЧНЫХ ГРУПП

Аманбай Аяулым, Даулетбек Улдана
karshygina84@mail.ru, d.matin@mai.ru
КарГУ им. Е.А. Букетова факультет математики и информационных технологий студент 1 курса специальности «Математика и информатика»,
Караганда, Казахстан
ЕНУ им. Л.Н. Гумилева, Механико-математический факультет
студент 1-го курса специальности «Информационных систем»
Нур-Султан, Казахстан
Научные руководители – Матин Даурен Тюлютаевич,
Каршыгина Гульден Жумабеккызы

В теории представлений рассматриваются конкретные реализации аксиоматических систем абстрактной алгебры. Эта теория берет начало в изучении групп подстановок и матричных