

Gimnazija Vič

Pythonov modul – Polinomi

Ime in priimek: Mentor:  
Jakob Jaš Klemen Bajec

Table of Contents

[1 Uvod 3](#_Toc195446615)

[2 Operediltev problema 3](#_Toc195446616)

[3 Orodja in tehnologije 3](#_Toc195446617)

[3.1 Razredi in »dunder« metode 3](#_Toc195446618)

[3.1.1 Primer uporabe »dunder metod« 3](#_Toc195446619)

[3.2 Uporaba modula 4](#_Toc195446620)

[3.2.1 Uvažanje modula 4](#_Toc195446621)

[3.2.2 Uporaba modula 5](#_Toc195446622)

[4 Opis delovanja 5](#_Toc195446623)

[4.1 Zgradba modula 5](#_Toc195446624)

[4.2 Konstruktor 6](#_Toc195446625)

[4.3 Deljenje 6](#_Toc195446626)

[4.3.1 Matematični opis deljenja 6](#_Toc195446627)

[4.3.2 Implementacija 7](#_Toc195446628)

[4.4 Druge metode 7](#_Toc195446629)

[4.4.1 Nove funkcije 8](#_Toc195446630)

[5 Primer uporabe 8](#_Toc195446631)

[6 Zaključek 9](#_Toc195446632)

[7 Viri 9](#_Toc195446633)

# Uvod

Polinomi imajo že tisočletja v matematiki temeljno vlogo. Kot izrazi, sestavljeni iz spremenljivk in konstant z uporabo osnovne aritmetike in nenegativnih celih eksponentov, so konceptualno preprosti in hkrati zelo uporabni. Njihove prve sledi segajo v babilonsko matematiko okoli leta 2000 pred našim štetjem, kjer so kvadratne probleme reševali geometrijsko [2].

V islamski zlati dobi so učenjaki, kot je al-Khwarizmi, formalizirali metode za reševanje enačb in postavili temelje za simbolno algebro. Sam izraz „algebra“ izhaja iz njegovega dela al-jabr. Kasneje so evropski matematiki, kot sta bila Viète in Descartes, uvedli simbolični zapis, ki je omogočil, da smo polinome obravnavali kot abstraktne objekte, in jih povezal z geometrijo prek kartezične ravnine [1].

V 19. stoletju so prizadevanja za razumevanje polinomskih enačb privedla do rojstva teorije grup z delom Galoisa in Abela [1]. Danes so polinomi še vedno bistveni v numeričnih metodah, računalništvu, kriptografiji in strojnem učenju.

Kljub svoji preprostosti polinomi še naprej ohranjajo stoletja matematičnih idej in misli, povezujoč starodavna odkritja s sodobnimi problemi.

# Operediltev problema

Programski jeziki sami po sebi nimajo vgrajene nobene sposobnosti obravnavanja polinomov. Zato moramo vsakič, ko jih potrebujemo, polinome implementirati sami.

Ker pa je to zelo časovno neučinkovito bi bila veliko boljša rešitev v naprej pripravljen program, ki ga lahko vključimo v naš projekt ter tako brez nepotrebnega dela operiramo z polinomi na enostavno izvedljiv in berljiv način.

# Orodja in tehnologije

## Razredi in »dunder« metode

Python uporabnikom omogoča prosto definiranje razredov in njihovih podrejenih metod. Med njimi so tudi t.i. »dunder« metode, s katerimi so definirane osnovne lastnosti in operacije vsakega razreda (npr. Uporaba funkcije print(); operatorjev kot +, -, \*, ...; pretvorbo razreda v druge podatkovne tipe ...) .

### Primer uporabe »dunder metod«

»Dunder« metodo *\_\_sub\_\_* program pokliče, ko je spremeljivka tega razreda znotraj programa prisotna zraven operaterja »-« (odštevanje). Vrednost, ki jo ta metoda vrne program vzame za rezultat odštevanja.

A computer screen shot of colorful text

AI-generated content may be incorrect.

Slika 1: Uporaba "dunder" metode \_\_sub\_\_

V tem primeru funkcija sprejme 2 polinoma. V primeru da je drugi polinom le število, ga odšteje od zadnjega koeficienta prvega polinoma. V primeru, da pa je drugi polinom res polinom njegovim koeficientom spremeni predznak in ga prišteje prvemu polinomu (prištevanje obratne vrednosti).

## Uporaba modula

### Uvažanje modula

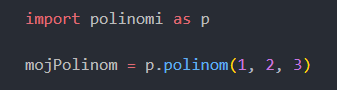
Python omogoča enostavno uvažanje modulov v svoje programe.

Najprej je potrebno datoteko, ki vsebuje modul shraniti v mapo, kjer se nahaja naš program. Nato pa z uporabo ukaza »import« programu povemo ime datoteke, iz katere naj uvozi modul. Lahko mu tudi povemo, pod katerim novim imenom želimo dostopati do razreda v modulu z uporabo ukaza »as«:

A screenshot of a computer

AI-generated content may be incorrect.

Slika 2: Priprava na uporabo modula



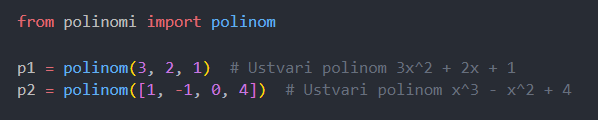
Slika 3: Uvoz modula v program

Uporaba ukaza »as« prepreči, da bi prišlo do napak pri uporabi, ki bi bile posledica enako imenovanih razredov in funkcij v modulu in v samem programu. To omogoča pythonov sistem »imenskega prostora«, ki vse funkcije, razrede, itd. modula polinomi kliče preko imena modula. Ker pa je pisanje ukaza »polinomi« zamudno, lahko z ukazom »as« to skrajšamo le v »p«. Tako lahko do razreda dostopamo preko ukaza: »p.polinom()«.

### Uporaba modula

#### Ustvarjanje polinoma

Polinom lahko usvarimo tako, da podamo njegove koeficiente ali v obliki seznama, ali pa kot posamezne paramtere.



Slika 4: Ustvarjanje polinoma

#### Operiranje s polinomom

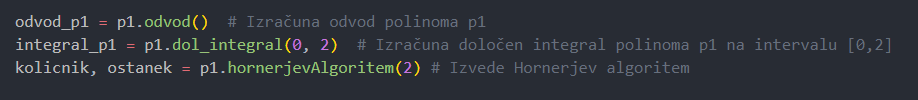
Za izvedbo aritmetičnih in logičnih operacij in indeksiranja se uporabljajo standardni pythonovi sistemi in operatorji

A screenshot of a computer code

AI-generated content may be incorrect.

Slika 5: Operiranje s polinomi

Za dodatno funkcionalnost pa so vgrajene tudi posebne funkcije.



Slika 6: Operiranje s polinomi

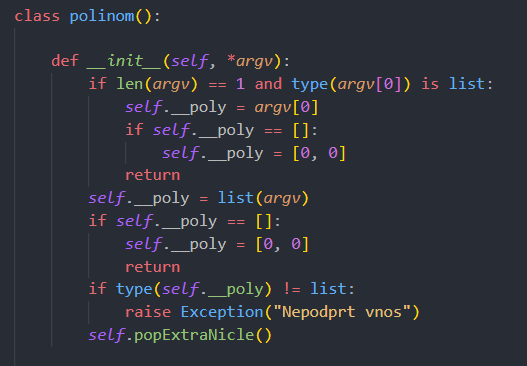
# Opis delovanja

## Zgradba modula

Modul vsebuje razred, imenovan »polinom«, ki omogoča delo s polinomi na podoben način, kot z ostalimi številskimi podakovnimi tipi. Razred vsebuje več metod, ki omogočajo osnovne operacije s polinomi. Koeficienti polinoma s shranjeni v seznamu, imovanem poly, ki je nedostopen izven razreda. Koeficienti si sredijo od leve proti desni, kjer koeficient *k* na *i*-tem mestu predstavlja koeficient istoležnega člena v polinomu.

## Konstruktor

Objekt razreda polinom se lahko ustvari s podajanjem posameznih koeficientov, ali pa s podajanjem seznama, ki vsebuje koeficiente. Program preveri vnos in zagotovi odstranitev vodilnih ničel.



Slika 7: Konstruktor

## Deljenje

Ena najbolj zapletenih operacij je deljenje polinomov.

### Matematični opis deljenja

Polinome se deli tako, da se najprej primerja vodilna člena. Če je vodilni člen deljenca višje stopnje kot od delitelja, se vodilna člena deli. Rezultat tega deljenja predstavlja vodilni člen količnika. Ta, pravkar dobljeni člen, pomnožimo z deliteljem in zmnožek odštejemo od deljenca. Nato ta postopek ponovimo z rezultatom tega odštevanja, dokler vodilni člen razlike ne bo nižje stopnje od delitelja. Če je razlika takrat neničelna to predstavlja ostanek.

Primer deljenja:

Količnik je v tem primeru torej: , ostanek pa: .

### Implementacija

Program najprej preveri, če sta in deljenec in količnik oba tipa polinom. V primeru, da nista, vrne napako. Nato preveri, da je stopnja deljenca višja od delitelja. V primeru, da ni, program prav tako vrne napako. Program nato izvede deljenje po postopku deljenja polonomov.

A screen shot of a computer program

AI-generated content may be incorrect.

Slika 8: Implementacija deljenja polinomov

## Druge metode

Razred ima poleg konstruktorja implementiranih še 26 drugih metod. Med njimi so metode za izpis polinoma (\_\_str\_\_, \_\_repr\_\_, raw), aritmetične operacije (\_\_add\_\_, \_\_radd\_\_, \_\_sub\_\_, \_\_rsub\_\_, \_\_mul\_\_, \_\_rmul\_\_, \_\_truediv\_\_, lshift), indeksiranje (\_\_getitem\_\_, \_\_setitem\_\_, \_\_delitem\_\_, pop), logične operacije (\_\_eq\_\_, \_\_ne\_\_) in drugo (stopnja, odvod, določeni integral, copy, hornerjev algoritem, vrednost, popExtraNicle, rshift).

### Nove funkcije

Poleg že vgrajenih metod so na novo dodane tudi sledeče:

* stopnja (vrne stopnjo polinoma)
* odvod (vrne polinom, ki je odvod tega polinoma)
* dolIntegral (vrne vrednost določenega integala)
* copy (kopira koeficiente drugega polinoma v tega)
* hornerjevAlgoritem (izvede hornerjev algoritem in vrne količnih in ostanek)
* vrednost (vrne vrednost polinoma za določen x)
* popExtraNicle (izbriše vodilne ničelne koeficiente iz seznama)
* lshift (množi polinom z )

A screen shot of a computer code

AI-generated content may be incorrect.

Slika 9: Nove funkcije

# Primer uporabe

Primer uporabe modula je funkcija, ki sprejme dva polinoma, in v primeru, da je prvi deljiv z drugim, polinoma pomnoži in deli z odvodom prvega polinoma ter izpiše količnik in ostanek tega deljenja.

A computer screen shot of a program code

AI-generated content may be incorrect.

Slika 10:Primer

Funkcija za dana polinoma a in b izpiše sledeče.

|  |  |
| --- | --- |
| A screen shot of a computer code  AI-generated content may be incorrect.  Slika 11: Izvedba primera | Slika 12: Izpis primera |

# Zaključek

Implementacija polinomov v pythonu ponuja zanesljivo in enostavno orodje, ki lahko služi kot osnova za nadaljnji razvoj aplikacij, ali pa kot orodje za delovanje z polinomi.

Njegova struktura omogoča enostavno izvajanje matematičnih operacij, od osnovnega vrednotenja polinoma do naprednejših metod, kot sta Hornerjev algoritem in določeni integral.

Zaradi svoje prilagodljivosti in intuitivne uporabe je modul koristen tako za študente kot tudi za razvijalce, ki potrebujejo učinkovito obdelavo polinomov v svojih programih. Z modularno zasnovo in jasnimi metodami omogoča razširljivost ter prilagoditev glede na potrebe uporabnika.

# Viri

[1] Boyer, C. B., & Merzbach, U. C. *A History of Mathematics* Wiley. (Datum zadnjega popravljanja 11.1. 2011). Dostopno na povezavi:  
<https://atiekubaidillah.wordpress.com/wp-content/uploads/2013/03/a-history-of-mathematics-3rded.pdf>

[2] Katz, V. J. *A History of Mathematics: An Introduction*. Addison-Wesley. (Datum zadnjega popravljanja 2009). Dostopno na povezavi: https://archive.org/details/a-history-of-mathematics-an-introduction-victor-j.-katz/page/n5/mode/2up

[3] PythonMorsels. Every dunder method in Python. (Datum zadnjega popravljanja: 19. 3. 2024). Dostopno na povezavi: https://www.pythonmorsels.com/every-dunder-method/