

PYTHON FUNKSIYALARIDAN FOYDALANIB ODDIY KALKULYATOR YARATISH

Mirsaid Yusupov Abdullaziz o'g'li

Farg'ona davlat universiteti

Amaliy matematika va informatika

kafedrası katta o'qituvchisi

E-mail: mirsaidbeky@gmail.com

Farg'ona davlat universiteti

Fizika va matematika fakulteti

Amaliy matematika yo'nalishi talabasi

Axmadaliyev Shuxratjon Xamdamjon o'g'li

E-mail: shuxratjonaxmadaliyev15@gmail.com

ANONATSIYA (Uzbek tilida)

Mazkur maqolada Python dasturlash tilida funksiyalar yordamida oddiy kalkulyator yaratish jarayoni tahlil qilinadi. Funksiyalarni modul tarzida tashkil etish, arifmetik amallarni alohida bloklarga ajratish, foydalanuvchi bilan muloqot jarayonini soddalashtirish va kodni qayta ishlatish imkoniyatlari yoritilgan. Shuningdek, kalkulyatorning algoritmik tuzilishi, funksional modullanishi hamda raqamli ta'lim jarayonida mazkur yondashuvning ahamiyati ilmiy asoslangan. Tadqiqot natijalari Python dasturlashni o'rganuvchilar uchun samarali amaliy misol sifatida xizmat qiladi.

Kalit so'zlar(o'zbek tilida): Python , Funksiya , Oddiy kalkulyator

АННОТАЦИЯ (rus tilida)

В данной статье анализируется процесс создания простого калькулятора с использованием функций в языке программирования Python. Подчеркиваются возможности модульной организации функций, разделения арифметических операций на отдельные блоки, упрощения процесса взаимодействия с пользователем и повторного использования кода. Также научно обосновываются алгоритмическая структура калькулятора, функциональная модульность и важность такого подхода в процессе цифрового образования. Результаты исследования служат эффективным практическим примером для изучающих программирование на Python.

Ключевые слова (rus tilida): Python, Функция, Простой калькулятор

ABSTRACT (ingiliz tili)

This article analyzes the process of creating a simple calculator using functions in the Python programming language. The possibilities of organizing functions in a modular way, separating arithmetic operations into separate blocks, simplifying the process of user interaction, and reusing code are highlighted. The algorithmic structure

of the calculator, functional modularization, and the importance of this approach in the process of digital education are also scientifically substantiated. The results of the research serve as an effective practical example for those learning Python programming.

Keywords (ingiliz tili): Python , Function , Simple calculator

XXI asr raqamli transformatsiya jarayonlarida dasturlash tillarining roli keskin oshdi. Ulardan biri — Python dasturlash tili — o'zining soddaligi, yuqori darajada o'qilishi va keng kutubxonalar tizimi bilan keng miqyosda qo'llanilmoqda. Ta'lim jarayonida Python tilidan foydalanish o'quvchilarga algoritmik tafakkurni shakllantirish, muloqotga kirishuvchi dasturlar yaratish hamda real tizimlar modellarini tuzishni o'rgatadi. Pythonning yana bir muhim ustunligi — funksional dasturlash elementlari bilan integratsiya qila olishi. Funksiyalar yordamida kodni modullar shaklida tashkil etish, uni qayta ishlatish, testlash va yangilash osonlashadi. Dastlabki bosqichda o'rgatiladigan eng soddada amaliy loyihalardan biri — oddiy kalkulyator yaratish bo'lib, bu jarayon o'quvchilarning funksiyalar, shart operatorlari, kiritish-chiqarish amallari va xatoliklar bilan ishlashni mukammal o'zlashtirishiga yordam beradi. Maqolada Python funksiyalari asosida oddiy kalkulyator yaratishning metodik, nazariy va amaliy jihatlari chuqur yoritiladi. Ushbu tadqiqot Python tilida modulli arxitektura asosida dastur yaratish bo'yicha amaliy tajribani to'plovchi o'quv-uslubiy material sifatida xizmat qiladi.

Python dasturlash tilining funksional yondashuvdagi o'rni

Python dasturlash tili zamonaviy texnologik modellar, sun'iy intellekt, ma'lumotlar tahlili, avtomatlashtirilgan tizimlar va axborot texnologiyalarining aksariyat yo'nalishlarida yetakchi o'rin tutadi. Ushbu tilning mashhurliigi, avvalo, uning sintaktik soddaligi, funksional imkoniyatlari, dinamik tiplanganligi, keng kutubxona bazasi va platformalardan mustaqil ishlash xususiyati bilan belgilanadi. Pythonning funksional dasturlash paradigmasi orqali muammolarni bloklarga ajratgan holda modulli yondashuv asosida yechish imkoniyati mavjud bo'lib, bu esa yangilanadigan, sinovdan oson o'tadigan va qayta foydalaniladigan kodlar yaratish imkonini beradi.

Funksiyalar — bu Python tilining asosiy konstruktiv elementlaridan biri bo'lib, ular dasturiy jarayonni lo'nda, izchil va boshqaruvchan qilishga imkon beradi. Funksiyalar parametrlar qabul qiladi, algoritmik amallarni bajaradi va natijani qaytaradi. Ushbu jarayon strukturaviy dasturlashning deterministik tamoyillari bilan bevosita bog'liq.

Oddiy kalkulyator yaratishning ilmiy-nazariy asoslari

Kalkulyator — matematik amallarni bajaruvchi eng sodda, ammo metodik jihatdan juda samarali dasturiy modullardan biridir. Ta'lim jarayonida oddiy kalkulyator yaratish o'quvchilarda quyidagi ko'nikmalarni shakllantiradi:

- abstrakt fikrlashni rivojlantirish;
- algoritmnlarni bosqichma-bosqich rejalashtirish;
- shart operatorlari bilan ishlash;
- funksiyalar o'rtasidagi bog'liqlikni tushunish;
- xatoliklardan himoyalalanish mexanizmlarini o'rganish.

Kalkulyator dasturining asosiy matematik modeli arifmetik amallardan iborat bo'lgani uchun, u orqali algoritmnlarning bajarilish tartibini, boshqaruv oqimini, tarmoqlanuvchi (branching) dastur tuzilishini ham amalda o'rganish mumkin. Bundan tashqari, kalkulyator yaratish jarayonida o'quvchi ma'lumot turlari, o'zgaruvchilar, operatorlar, foydalanuvchi bilan muloqot interfeysi (CLI) kabi tushunchalarni puxta o'zlashtiradi.

Kalkulyatorning mantiqiy-arxitektura modeli

Oddiy kalkulyator funksional bloklardan tashkil topadi. Har bir blokning vazifasi aniq belgilangan:

1. **Kirish bloki** — foydalanuvchidan sonlar va amal turini qabul qiladi.
2. **Tahlil va boshqaruv bloki** — amal turini aniqlaydi, mos funksiyani ishga tushiradi.
3. **Arifmetik funksiyalar bloki** — qo'shish, ayirish, ko'paytirish, bo'lish.
4. **Xatoliklarni qayta ishlash bloki** — nolga bo'lish va noto'g'ri amal kiritilgan hollarni aniqlaydi.
5. **Chiqish bloki** — final natijani foydalanuvchiga bildiradi.

Mazkur bloklarning funksional o'zaro ta'siri algoritmik strukturani shakllantiradi. Bu struktura dasturiy injiniringning modularity, clarity, reusability tamoyillariga mos keladi.

Python funksiyalaridan foydalanib oddiy kalkulyatorni amalga oshirish

Oddiy kalkulyator dasturini Python dasturlash tilida yaratishda funksional yondashuv asos qilib olinadi. Har bir arifmetik amal alohida funksiya ko'rinishida tashkil etiladi, bu esa kodning o'qilishi, tushunilishi va qayta ishlatilishini sezilarli darajada oshiradi. Quyida oddiy kalkulyatorning asosiy funksional komponentlari keltirilgan.

```
def qoshish(a, b):
```

```
    return a + b
```

```
def ayirish(a, b):
```

```
    return a - b
```

```
def kopaytirish(a, b):
```

```
    return a * b
```

```
def bolish(a, b):
    if b == 0:
        return "Xatolik: nolga bo'lish mumkin emas"
    return a / b
```

Mazkur funksiyalar arifmetik amallarni mustaqil bloklar sifatida bajaradi. Har bir funksiya ikki parametr qabul qiladi va mos matematik natijani qaytaradi. Bo'lish funksiyasida nolga bo'lish holatini aniqlash uchun shart operatoridan foydalanilgan.

Foydalanuvchi bilan muloqotni tashkil etish

Kalkulyatorning muhim jihatlaridan biri foydalanuvchi bilan samarali muloqot qilishidir. Ushbu jarayonda kiritish-chiqarish operatorlari yordamida foydalanuvchidan ma'lumotlar olinadi va natija ekranga chiqarilad

```
print("Oddiy kalkulyator")
print("Amallar: + - * /")
amal = input("Amalni tanlang: ")
son1 = float(input("Birinchi sonni kiriting: "))
son2 = float(input("Ikkinchi sonni kiriting: "))
```

```
if amal == "+":
    natija = qoshish(son1, son2)
elif amal == "-":
    natija = ayirish(son1, son2)
elif amal == "*":
    natija = kopaytirish(son1, son2)
elif amal == "/":
    natija = bolish(son1, son2)
else:
    natija = "Noto'g'ri amal tanlandi"
```

```
print("Natija:", natija)
```

Mazkur kodda shart operatorlari yordamida foydalanuvchi tanlagan amal aniqlanadi va mos funksiya chaqiriladi. Bu esa dasturda boshqaruv oqimini aniq va tushunarli tashkil etishga xizmat qiladi.

Dastur samaradorligi va pedagogik ahamiyati

Funksiyalar asosida yaratilgan oddiy kalkulyator dasturi quyidagi pedagogik va texnik afzalliklarga ega:

1. kodni modullarga ajratish orqali mantiqiy tuzilma shakllanadi;
2. funksiyalar yordamida dastur qismlari qayta ishlatiladi;

3. xatoliklarni aniqlash va oldini olish osonlashadi;

4. o'quvchilarda algoritmik fikrlash va strukturaviy tahlil ko'nikmalari rivojlanadi.

Mazkur yondashuv boshlang'ich darajadagi dasturchilar uchun Python tilining asosiy imkoniyatlarini amaliy misol orqali mustahkamlash imkonini beradi.

Ilova :

```
import tkinter as tk
import math
```

```
def bos(son):
    display.insert(tk.END, son)
```

```
def tozalash():
    display.delete(0, tk.END)
```

```
def hisobla():
    try:
        matn = display.get()

        if matn.endswith("!"):
            son = int(matn[:-1])
            natija = math.factorial(son)
        else:
            natija = eval(matn)
```

```
    display.delete(0, tk.END)
    display.insert(0, natija)
```

```
except:
    display.delete(0, tk.END)
    display.insert(0, "Xatolik")
```

```
oyna = tk.Tk()
oyna.title("Oddiy Kalkulyator")
oyna.geometry("420x400")
oyna.configure(bg="blue")
oyna.resizable(False, False)
```

```
display = tk.Entry(
    oyna,
    font=("Arial", 18),
    justify="right",
    bd=5
)
display.grid(row=0, column=0, columnspan=5, padx=10, pady=10, ipady=10)
```

```
tugmalar = [
    ('1',1,0), ('2',1,1), ('3',1,2), ('*',1,3), ('/',1,4),
    ('4',2,0), ('5',2,1), ('6',2,2), ('+',2,3), ('-',2,4),
    ('7',3,0), ('8',3,1), ('9',3,2), ('**',3,3), ('!',3,4),
    ('.',4,0), ('0',4,1), ('%',4,2), ('=',4,3), ('C',4,4),
]
```

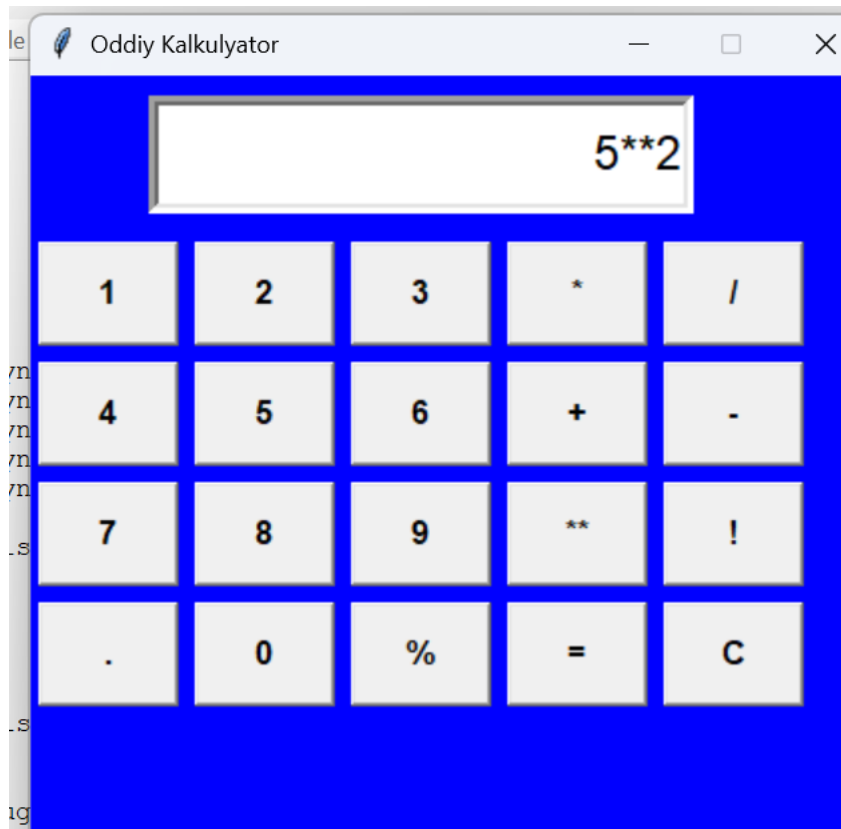
```
for (text, r, c) in tugmalar:
    if text == "=":
        cmd = hisobla
    elif text == "C":
        cmd = tozalash
    else:
        cmd = lambda t=text: bos(t)
```

```
tk.Button(
    oyna,
    text=text,
    width=6,
    height=2,
    font=("Arial", 12, "bold"),
    command=cmd
).grid(row=r+1, column=c, padx=4, pady=4)
```

```
def foiz():
    try:
        qiymat = float(display.get())
        display.delete(0, tk.END)
        display.insert(0, qiymat / 100)
    except:
        display.delete(0, tk.END)
```

```
display.insert(0, "Xatolik")
```

```
oyna.mainloop()
```



Ilovaning ishlash jarayoni

Taqdim etilgan Python dasturi foydalanuvchiga grafik interfeys orqali arifmetik amallarni bajarish imkonini beradi. Kalkulyatorning asosiy ishlash jarayoni quyidagi bosqichlardan iborat:

1. **Foydalanuvchi bilan muloqot** – tkinter kutubxonasi yordamida oyna va tugmalar interfeysi yaratiladi. Foydalanuvchi raqamlar va amallarni tanlab kiritadi.
2. **Ma'lumotni qabul qilish** – foydalanuvchi tanlagan amal va raqamlar Entry widgeti orqali olinadi.
3. **Boshqaruv va hisoblash** – foydalanuvchi tanlagan amal turiga qarab, mos funksiya chaqiriladi. Masalan, + amalida qoshish(), factorial uchun esa math.factorial() ishlatiladi. Shu bilan birga, shart operatorlari xatoliklarni aniqlash va oldini olish imkonini beradi.
4. **Natijani chiqarish** – hisoblangan natija qayta Entry widgetida foydalanuvchiga ko'rsatiladi. Bu jarayon interaktiv va intuitiv tarzda amalga oshiriladi.

Dastur xatoliklarni aniqlash mexanizmi bilan jihozlangan bo'lib, foydalanuvchi noto'g'ri amal yoki nolga bo'lish kiritgan hollarda tegishli ogohlantirish beradi. Shu tariqa, dastur ishonchli va pedagogik jihatdan samarali tarzda ishlaydi.

Ilovaning kelajakdagi qo'llanilishi

Ushbu kalkulyator ilovasi nafaqat dasturlashni o'rganuvchilar uchun amaliy mashq sifatida, balki quyidagi sohalarda ham keng qo'llanilishi mumkin:

1. **Ta'lim jarayoni** – boshlang'ich va o'rta darajadagi dasturchilar uchun algoritmik fikrlashni rivojlantirish, funksiyalar va shart operatorlarini amaliy misolda o'rganish.
2. **Ma'lumotlarni hisoblash va avtomatlashtirish** – oddiy arifmetik operatsiyalarni bajaruvchi modul sifatida boshqa dasturiy loyihalarga integratsiya qilish.
3. **Interaktiv o'quv loyihalari** – GUI bilan ishlash ko'nikmalarini rivojlantirish, foydalanuvchi interfeysi dizayni va xatoliklarni qayta ishlash mexanizmini o'rganish.
4. **Kengaytirilgan funksiyalar qo'shish** – kelajakda kalkulyatorga murakkab matematik funksiyalar, grafik chizish imkoniyati yoki moliyaviy hisob-kitob modulini qo'shish orqali yanada rivojlantirish mumkin.

Shunday qilib, dastur ta'lim va amaliy loyihalar uchun mustahkam platforma bo'lib xizmat qiladi.

Xulosa

Ushbu maqolada Python dasturlash tilida funksiyalar yordamida oddiy kalkulyator yaratish jarayoni **nazariy va amaliy jihatlar bilan** yoritildi. Kalkulyatorning modul tuzilishi, foydalanuvchi bilan muloqot mexanizmi va xatoliklarni qayta ishlash bloklari ilmiy asoslangan tarzda taqdim etildi.

Natijalar:

Funksiyalar orqali kodni modulga ajratish, qayta ishlatish va testlash imkoniyati oshadi.

Bosqichma-bosqich algoritmik tafakkur shakllanadi.

O'quvchilar shart operatorlari, arifmetik amallar va foydalanuvchi interfeysini amaliy misolda o'rganadilar.

Ilova kelajakda interaktiv o'quv loyihalari, avtomatlashtirilgan hisob-kitob va murakkab dasturlarni yaratishda keng qo'llanilishi mumkin.

Shunday qilib, mazkur yondashuv Python dasturlashni o'rganuvchilar uchun nafaqat amaliy mashq, balki ilmiy va pedagogik jihatdan foydali vosita sifatida xizmat qiladi.

Foydalanilgan adabiyotlar

1. Шолле Франсуа. Глубокое обучение на Python. 2-е межд. издание. — СПб.: Питер, 2023. — 576 с.
2. Sh. A.Mengliyev, O.A.Abdugʻaniev, S.Q.Shonazarov, D.Sh.Toʻrayev. Python dasturlash tili. — Termiz, 2021. — 156 с.
3. Anvar Narzullayev. Pythonda dasturlash asoslari. — T.: Akademnashr, 2021. — 336 s.
4. M.K.Onarkulov, I.N. Tojimamatov. Python dasturlash tili, Fargʻona, FarDU nusxa koʻpaytirish boʻlimi, 2023. - 178 с.
5. Guido van Rossum, Fred L. Drake Jr.. Python Programming Language. — O'Reilly Media, 2022. — 480 p.
6. Mark Lutz. Learning Python. — O'Reilly Media, 2021. — 1200
7. Shahriyozov, F. Python dasturlash tili asoslari. —Toshkent: “Fan va texnologiya”, 2021. —220 b.
8. <https://www.ziyonet.uz/> - Axborot taʼlim portal
9. <https://metanit.com/python/tutorial/>
10. <https://www.youtube.com/@Sariqdev> dasturlash boʻyicha video darsliklar mavjud..
11. <https://docs.python.org/3/library/tkinter.html>
12. <https://www.python.org/downloads/>