

AVTOMOBIL SINFI MISOLIDA SINFI METODLARINI YOZISH

Mirsaid Yusupov Abdulaziz o'g'li

Farg'ona davlat universiteti Amaliy matematika

va informatika kafedrasi o'qituvchisi

E-mail: mirsaidbeky@gmail.com

Karimova Noila Zafarjon qizi

Farg'ona davlat universiteti Amaliy matematika

yo'nalishi 3-bosqich 23.07-guruh talabasi

E-mail: noilakarimova86@gmail.com

Annotatsiya: Ushbu maqola Python dasturlash tilida ob'ektga yo'naltirilgan dasturlashda sinfi metodlarini avtomobil sinfi amaliy misoli asosida keng qamrovli tadqiq etadi. Ishda ob'ektga yo'naltirilgan yondashuvning nazariy asoslari, sinflar va ob'ektlar kontseptsiyasi, shuningdek sinfi metodlarining turli turlari batafsil ko'rib chiqiladi. Ekzemplar metodlari, sinfi metodlari va statik metodlarni, ularning sintaktik xususiyatlari va amaliy qo'llanilishini tahlil qilish alohida e'tiborga olingan. Tadqiqot avtomobil sinfini ishlab chiqish kontekstida inkapsulyatsiya, meros va polimorfizm tamoyillarini qamrab oladi. Python ning maxsus metodlari, jumladan konstruktorlar, destruktorlar va sehrli metodlar batafsil tahlil qilinadi. Sinf metodlarini yozishning dizayn patternlari va eng yaxshi amaliyotlari ko'rib chiqiladi. Maqola avtomobil holatini boshqarish uchun turli xil metodlarni amalga oshirishning amaliy misollarini o'z ichiga oladi, jumladan dvigatelni ishga tushirish, tezlikni boshqarish, texnik xizmat ko'rsatish va diagnostika metodlari. Sinf metodlarini tashkil qilishning turli yondashuvlarining afzalliklari va kamchiliklari tahlil qilinadi. Ish kodni hujjatlashtirish, metodlarni sinovdan o'tkazish va PEP 8 kodlash standartlariga rioya qilish bo'yicha tavsiyalarni o'z ichiga oladi. Tadqiqot sinfi metodlarini to'g'ri loyihalash ishonchli, miqyoslanadigan va qo'llab-quvvatlanadigan dasturiy ta'minotni yaratishga qanday hissa qo'shishini ko'rsatadi.

Kalit so'zlar: ob'ektga yo'naltirilgan dasturlash, Python, sinfi metodlari, ekzemplar metodlari, statik metodlar, inkapsulyatsiya, meros, polimorfizm, avtomobil sinfi, sehrli metodlar

Аннотация: Данная статья представляет комплексное исследование методов классов в объектно-ориентированном программировании на языке Python с использованием практического примера класса автомобиля. В работе детально рассматриваются теоретические основы объектно-ориентированного подхода, концепция классов и объектов, а также различные типы методов класса. Особое внимание уделяется анализу экземплярных методов, методов класса и статических методов, их синтаксическим особенностям и практическому

применению. Исследование охватывает принципы инкапсуляции, наследования и полиморфизма в контексте разработки класса автомобиля. Подробно анализируются специальные методы Python, включая конструкторы, деструкторы и магические методы. Рассматриваются паттерны проектирования и лучшие практики написания методов классов. Статья содержит практические примеры реализации различных методов для управления состоянием автомобиля, включая методы для запуска двигателя, управления скоростью, обслуживания и диагностики. Представлен анализ преимуществ и недостатков различных подходов к организации методов класса. Работа включает рекомендации по документированию кода, тестированию методов и соблюдению стандартов кодирования PEP 8. Исследование демонстрирует, как правильное проектирование методов класса способствует созданию надежного, масштабируемого и поддерживаемого программного обеспечения.

Ключевые слова: объектно-ориентированное программирование, Python, методы класса, экземплярные методы, статические методы, инкапсуляция, наследование, полиморфизм, класс автомобиля, магические методы

Annotation: This article presents a comprehensive study of class methods in object-oriented programming using Python, with a practical example of a car class. The work thoroughly examines the theoretical foundations of the object-oriented approach, the concept of classes and objects, and various types of class methods. Special attention is given to analyzing instance methods, class methods, and static methods, their syntactic features, and practical applications. The research covers the principles of encapsulation, inheritance, and polymorphism in the context of developing a car class. Special Python methods are analyzed in detail, including constructors, destructors, and magic methods. Design patterns and best practices for writing class methods are examined. The article contains practical examples of implementing various methods for managing car state, including methods for starting the engine, controlling speed, maintenance, and diagnostics. An analysis of advantages and disadvantages of different approaches to organizing class methods is presented. The work includes recommendations for code documentation, method testing, and adherence to PEP 8 coding standards. The research demonstrates how proper design of class methods contributes to creating reliable, scalable, and maintainable software.

Keywords: object-oriented programming, Python, class methods, instance methods, static methods, encapsulation, inheritance, polymorphism, car class, magic methods

KIRISH

Ob'ektga yo'naltirilgan dasturlash zamonaviy dasturiy ta'minot ishlab chiqishning asosiy paradigmalaridan biri hisoblanadi. Bu yondashuv haqiqiy dunyo

ob'ektlarini dasturlashda modellashtirish imkonini beradi va murakkab tizimlarni modulli, qayta ishlatiluvchi va tushunarli tarzda tashkil etishga yordam beradi. Python dasturlash tili ob'ektga yo'naltirilgan dasturlashni to'liq qo'llab-quvvatlaydi va kuchli sinf tizimiga ega.

Sinflar ob'ektlarni yaratish uchun shablon yoki loyiha bo'lib xizmat qiladi. Har bir sinf ma'lumotlarni saqlash uchun atributlar va bu ma'lumotlar bilan ishlash uchun metodlarga ega. Metodlar sinfning xatti-harakatini belgilaydi va ob'ektlarning funksionalligini ta'minlaydi. Python tilida uch asosiy metod turi mavjud: ekzemplar metodlari, sinf metodlari va statik metodlar.

Avtomobil sinfi ob'ektga yo'naltirilgan dasturlashning tamoyillarini o'rganish uchun klassik va tushunarli misoldir. Avtomobil haqiqiy dunyo ob'ekti sifatida ko'plab xususiyatlarga va xatti-harakatlarga ega. Marka, model, rang, yil, tezlik kabi atributlar va dvigatelni ishga tushirish, tezlikni oshirish, to'xtash kabi metodlar avtomobilning asosiy xususiyatlarini aks ettiradi.

Sinf metodlarini to'g'ri yozish dasturiy ta'minotning sifatiga bevosita ta'sir qiladi. Yaxshi tashkil etilgan metodlar kodni o'qish va tushunishni osonlashtiradi, xatolarni kamaytirishga yordam beradi va dasturni kengaytirishni soddalashtiradi. Metodlarni loyihalashda inkapsulyatsiya, abstraksiya, meros va polimorfizm kabi ob'ektga yo'naltirilgan dasturlashning asosiy tamoyillariga rioya qilish muhimdir.

Ushbu tadqiqot ishining maqsadi Python dasturlash tilida avtomobil sinfi misolida sinf metodlarini yozishning nazariy asoslari va amaliy jihatlarini chuqur o'rganishdir. Maqolada turli turdagi metodlar, ularning sintaksisi, qo'llanilish sohalari va eng yaxshi amaliyotlar batafsil ko'rib chiqiladi. Tadqiqot natijalarida dasturchilar sinf metodlarini samarali loyihalash va amalga oshirish bo'yicha amaliy bilim va ko'nikmalar oladi.

Asosiy Qism

Ob'ektga yo'naltirilgan dasturlash paradigmasi to'rt asosiy tamoyilga asoslanadi: inkapsulyatsiya, meros, polimorfizm va abstraksiya. Bu tamoyillar dasturiy ta'minotni tashkil etishning kuchli va moslashuvchan usulini taqdim etadi.

Inkapsulyatsiya ma'lumotlar va ular bilan ishlovchi metodlarni bitta birlik ichida birlashtirish tamoyilidir. Bu tamoyil ichki implementatsiya tafsilotlarini yashirish va faqat zarur interfeys orqali ob'ekt bilan o'zaro ta'sirni ta'minlash imkonini beradi. Python tilida inkapsulyatsiya nomi bilan belgilanadigan public, protected va private atributlar va metodlar orqali amalga oshiriladi.

Public atributlar va metodlar hech qanday cheklovsiz istalgan joydan kirish mumkin bo'lgan elementlardir. Ular oddiy nomlar bilan aniqlanadi va sinfning ochiq interfeysini tashkil etadi. Protected atributlar bitta pastki chiziq bilan boshlanadi va ular sinfdan va uning vorislaridan kirish mumkin bo'lgan elementlarni bildiradi. Private

atributlar ikki pastki chiziq bilan boshlanadi va ular faqat sinf ichida ishlatilishi mo'ljallangan.

Meros bir sinfning boshqa sinfdan xususiyatlar va metodlarni meros olish mexanizmidir. Bu kod qayta ishlatilishini ta'minlaydi va sinflar ierarxiyasini yaratish imkonini beradi. Python tilida bir nechta meros ham qo'llab-quvvatlanadi, bu bir sinfning bir nechta ota sinflardan meros olishiga imkon beradi.

Polimorfizm turli sinflarning ob'ektlari bir xil interfeys orqali ishlatilishi mumkinligini anglatadi. Bu har xil ob'ektlar bir xil metod nomlaridan foydalanishi mumkin bo'lgan holda, har bir ob'ekt o'z implementatsiyasiga ega bo'lishini ta'minlaydi. Python tilida polimorfizm dinamik tipizatsiya orqali tabiiy tarzda qo'llab-quvvatlanadi.

Abstraksiya muhim xususiyatlarni ajratib olish va ahamiyatsiz tafsilotlarni yashirish jarayonidir. Abstrakt sinflar va interfeys tushunchalari abstraktsiyani amalga oshirish uchun ishlatiladi. Python tilida ABC moduli abstrakt bazaviy sinflarni yaratish imkonini beradi.

Sinf ob'ektlarni yaratish uchun shablon hisoblanadi. U atributlar va metodlar to'plamini belgilaydi. Ob'ekt sinfning konkret namunasi yoki ekzemplari hisoblanadi. Har bir ob'ekt o'z holatiga ega bo'ladi, lekin barcha ob'ektlar sinf tomonidan belgilangan bir xil tuzilish va xatti-harakatlarga ega.

Python tilida sinf yaratish uchun class kalit so'zidan foydalaniladi. Sinf nomi odatda bosh harf bilan boshlanadi va CamelCase konvensiyasiga amal qiladi. Sinf tanasi ichki chiziq bilan ajratilgan blokda joylashadi va barcha sinf a'zolari shu blok ichida aniqlanadi.

Konstruktor yoki initsializator sinf ekzemplari yaratilganda avtomatik ravishda chaqiriladigan maxsus metoddir. Python tilida konstruktor init nomi bilan aniqlanadi. Bu metod yangi yaratilgan ob'ektning boshlang'ich holatini o'rnatish uchun ishlatiladi. Konstruktor birinchi parametr sifatida self ni qabul qiladi, bu joriy ob'ektga havola hisoblanadi.

Self parametri Python tilida sinfning barcha ekzemplar metodlarida majburiy birinchi parametr hisoblanadi. U joriy ob'ektga murojaat qilish imkonini beradi va ob'ektning atributlari va metodlariga kirish uchun ishlatiladi. Self nomi konvensiya bo'lib, nazariy jihatdan istalgan nom ishlatilishi mumkin, lekin Python jamoasi self nomini qat'iy tavsiya qiladi.

Ekzemplar metodlari ob'ektning holati bilan ishlovchi va uni o'zgartiradigan metodlardir. Ular birinchi parametr sifatida self ni qabul qiladi va ob'ektning atributlariga kirish huquqiga ega. Ekzemplar metodlari sinfning asosiy funktsionalligini ta'minlaydi va eng ko'p ishlatiladigan metod turidir.

Sinf metodlari sinfning o'zi bilan ishlovchi metodlardir. Ular classmethod dekoratori bilan belgilanadi va birinchi parametr sifatida cls ni qabul qiladi, bu sinf

ob'ektiga havola hisoblanadi. Sinf metodlari alternativ konstruktorlar yaratish yoki sinf darajasidagi operatsiyalarni bajarish uchun ishlatiladi.

Statik metodlar sinf yoki ob'ekt holatiga bog'liq bo'lmagan yordamchi funksiyalardir. Ular `staticmethod` dekoratori bilan belgilanadi va na `self` na `cls` parametrlarini qabul qilmaydi. Statik metodlar sinf nomlar fazosida mantiqiy jihatdan guruhlanishi kerak bo'lgan funksiyalar uchun foydalidir.

Avtomobil sinfini loyihalashda birinchi qadam uning asosiy atributlarini aniqlashdir. Avtomobil markasi, modeli, ishlab chiqarilgan yili, rangi, dvigatel hajmi, joriy tezligi, yurgan masofasi, yonilg'i miqdori kabi xususiyatlarga ega bo'lishi mumkin. Bu atributlar ob'ektning holatini ifodalaydi va konstruktorda initsializatsiya qilinadi.

Konstruktor metodida avtomobilning boshlang'ich holati o'rnatiladi. Ba'zi parametrlar majburiy bo'lishi mumkin, masalan marka va model, boshqalari esa standart qiymatlarga ega bo'lishi mumkin, masalan boshlang'ich tezlik nolga teng. Konstruktor parametrlarini to'g'ri tanlash sinfning foydalanishini osonlashtiradi va xatolar ehtimolini kamaytiradi.

Atributlarni `private` yoki `protected` sifatida belgilash inkapsulyatsiya tamoyilini amalga oshirishga yordam beradi. Masalan, dvigatelning ishlash holati yoki yonilg'i miqdori kabi muhim atributlar `private` bo'lishi mumkin. Bu ularning to'g'ridan-to'g'ri o'zgartirilishining oldini oladi va faqat maxsus metodlar orqali kirish imkonini beradi.

`Property` dekoratori atributlarga boshqariladigan kirish ta'minlash uchun ishlatiladi. U `getter`, `setter` va `deleter` metodlarini yaratish imkonini beradi. `Property` yordamida atributlarga oddiy kirish sintaksisi saqlanadi, lekin qo'shimcha mantiq qo'shilishi mumkin, masalan validatsiya yoki hisoblashlar.

`String` tasviri metodlari ob'ektni matn ko'rinishida ifodalash uchun ishlatiladi. `str` metodi foydalanuvchilarga tushunarli tavsif yaratadi, `repr` metodi esa ishlab chiquvchilar uchun aniq va to'liq tasvirni beradi. Bu metodlar `debug` qilish va `logging` jarayonlarida juda foydali.

Solishtirish metodlari ob'ektlarni bir-biri bilan solishtirish imkonini beradi. `eq`, `lt`, `gt` kabi metodlar avtomobillarni turli parametrlar bo'yicha solishtirish uchun ishlatilishi mumkin. Masalan, avtomobillarni yurgan masofasi yoki ishlab chiqarilgan yili bo'yicha solishtirish.

Avtomobil sinfida dvigatelni boshqarish uchun bir nechta asosiy metodlar kerak. Dvigatelni ishga tushirish metodi avtomobilning holatini tekshiradi va dvigatelni ishlatish imkonini beradi. Bu metod yonilg'i mavjudligini, dvigatelning texnik holatini va boshqa zarur shartlarni tekshirishi mumkin.

Dvigatelni ishga tushirish metodida mantiqiy tekshiruvlar amalga oshiriladi. Agar dvigatel allaqachon ishlab turgan bo'lsa, metod xabar qaytarishi yoki `exception`

yuzaga keltirishi mumkin. Agar barcha shartlar bajarilsa, dvigatelning holati o'zgartiriladi va muvaffaqiyatli xabar qaytariladi.

Dvigatelni o'chirish metodi xavfsiz to'xtatish jarayonini ta'minlaydi. Bu metod avtomobil harakatda emasligini tekshirishi kerak. Agar avtomobil tezlik bilan harakatlanayotgan bo'lsa, dvigatelni o'chirish xavfli bo'lishi mumkin va metod ogohlantirish beradi yoki amaliyotni rad etadi.

Dvigatel diagnostikasi metodi dvigatelning joriy holatini tahlil qiladi va ma'lumot qaytaradi. Bu metod dvigatel harorati, moy bosimi, xatolar kodi va boshqa muhim parametrlarni tekshirishi mumkin. Diagnostika metodlari odatda lug'at yoki maxsus ob'ekt ko'rinishida batafsil ma'lumot qaytaradi.

Yonilg'i iste'moli hisoblash metodi avtomobil yurgan masofaga va dvigatel parametrlariga asoslangan holda yonilg'i sarfini hisoblab beradi. Bu metod avtomobil samaradorligini baholash va yonilg'ini to'ldirish zaruratini aniqlash uchun foydalidir.

Exception handling dvigatel bilan bog'liq metodlarda muhim rol o'ynaydi. Turli xil xatolar yuzaga kelishi mumkin, masalan yonilg'i tugashi, dvigatelning qizib ketishi yoki texnik nosozliklar. Custom exception sinflarini yaratish va ularni to'g'ri ishlash xatolarni aniq boshqarish imkonini beradi.

Tezlikni oshirish metodi avtomobilning tezligini oshirish uchun ishlatiladi. Bu metod dvigatelning ishlash holatini, maksimal tezlik chegarasini va boshqa xavfsizlik parametrlarini tekshiradi. Tezlikni sekin oshirish yoki maksimal ruxsat etilgan tezlikka cheklash kabi mantiq qo'shilishi mumkin.

Tezlikni kamaytirish metodi sekinlashtirish va to'xtatish jarayonlarini boshqaradi. Bu metod tormozlash tizimining holatini tekshirishi va tezlikni xavfsiz tarzda kamaytirishi kerak. Keskin tormozlash hollari uchun maxsus mantiq qo'shilishi mumkin.

Cruise control funksiyasi avtomobilni doimiy tezlikda ushlab turish uchun ishlatiladi. Bu metod kerakli tezlikni o'rnatadi va avtomatik ravishda saqlaydi. Cruise control faollashtirilganda, tezlikni qo'lda o'zgartirish chegaralangan bo'lishi mumkin.

Yo'l sharoitlariga moslashish metodlari turli haydash rejimlarini amalga oshiradi. Sport rejimi, ekonomik rejim, qish rejimi kabi turli sharoitlar uchun dvigatel va tormoz tizimlarining xatti-harakatini o'zgartiradi. Bu metodlar avtomobilning moslashuvchanligini oshiradi.

GPS navigatsiya bilan integratsiya metodi avtomobilning joriy holatini va yo'l ma'lumotlarini qayta ishlaydi. Bu metod masofani hisoblash, kelish vaqtini bashorat qilish va optimal marshrutni tanlash funksiyalarini ta'minlashi mumkin.

Avtopilot yoki yordamchi haydash tizimlari uchun metodlar zamonaviy avtomobillarda muhim ahamiyatga ega. Bu metodlar sensorlar ma'lumotlarini qayta ishlaydi, yo'l belgislarini taniydi va xavfsiz harakatlanishni ta'minlaydi. Bunday

metodlar juda murakkab mantiqni o'z ichiga oladi va ko'p sonli yordamchi funksiyalarni talab qiladi.

Texnik xizmat ko'rsatish rejasi metodi avtomobilning yurgan masofasi va vaqtga asoslanib texnik xizmat zaruratini aniqlaydi. Bu metod keyingi texnik ko'rikni rejalashtirishga yordam beradi va kerakli xizmat turlarini tavsiya qiladi. Profilaktik xizmat jadvali avtomobilning uzoq muddatli ishlashini ta'minlaydi.

Yog' almashtirish metodi dvigatel yog'ining holatini tekshiradi va almashtirish vaqti kelganda signal beradi. Bu metod yurgan masofa, dvigatel ishlash soatlari va yog' sifati parametrlariga asoslanib qaror qabul qiladi. Yog' sifatini monitoring qilish dvigatel ishlashining uzoq umrini ta'minlaydi.

Shinalarni tekshirish metodi shina bosimi, protektori iste'moli va umumiy holatini tahlil qiladi. Bu metod xavfsizlik uchun juda muhim va shinalardagi muammolarni erta aniqlashga yordam beradi. Shina almashtirish tavsiyalari avtomatik ravishda berilishi mumkin.

Tormoz tizimini diagnostika qilish metodi tormoz kolodkalari, disklari va suyuqlik holatini tekshiradi. Tormoz tizimining ishlashi avtomobil xavfsizligida hal qiluvchi rol o'ynaydi. Metod tormoz samaradorligini o'lchaydi va zarur ta'mirlar haqida ma'lumot beradi.

Elektr tizimi diagnostikasi metodi batareya holati, generator ishlashi va elektr tizimidagi xatolarni aniqlaydi. Zamonaviy avtomobillarda elektr tizimlari juda murakkab va ko'plab komponentlarni boshqaradi. Elektr tizimi diagnostikasi avtomobilning ishonchli ishlashini ta'minlaydi.

Xatolar jurnalini saqlash va tahlil qilish metodi barcha yuzaga kelgan xatolar va diagnostika natijalarini yozib boradi. Bu ma'lumotlar texnik xizmat tarixini saqlash, muammolarni tahlil qilish va kelajakdagi nosozliklarni bashorat qilish uchun ishlatiladi. Xatolar jurnali JSON yoki boshqa formatda saqlanishi mumkin.

Sinf metodlari avtomobil sinfida turli maqsadlar uchun ishlatilishi mumkin. Masalan, muayyan parametrlarga ega bo'lgan avtomobilni yaratish uchun alternativ konstruktorlar sifatida sinf metodlaridan foydalanish mumkin. Elektr avtomobil, gibrid avtomobil yoki sport avtomobil uchun alohida sinf metodlari yaratilishi mumkin.

Factory metodlari dizayn patterni sifatida sinf metodlari orqali amalga oshiriladi. Bu metodlar turli xil konfiguratsiyalarda avtomobil ob'ektlarini yaratadi. Masalan, standart komplektatsiya, to'liq komplektatsiya yoki maxsus buyurtma bo'yicha avtomobil yaratish uchun alohida factory metodlari bo'lishi mumkin.

Sinf darajasidagi statistikani boshqarish uchun ham sinf metodlaridan foydalaniladi. Yaratilgan avtomobillar soni, o'rtacha tezlik, umumiy yurgan masofa kabi ma'lumotlar sinf o'zgaruvchilarida saqlanishi va sinf metodlari orqali qayta ishlanishi mumkin.

Statik metodlar yordamchi funksiyalar sifatida avtomobil sinfi bilan bog'liq, lekin konkret ob'ekt yoki sinf holatiga bog'liq bo'lmagan operatsiyalarni bajaradi. Masalan, yonilg'i iste'molini hisoblash, tezlikni bir o'lchov birligidan boshqasiga o'tkazish, masofani hisoblash kabi funksiyalar statik metodlar sifatida amalga oshirilishi mumkin.

Validatsiya funksiyalari ham statik metodlar sifatida yozilishi mumkin. Marka nomini, model yilini, rang kodini tekshirish kabi validatsiya mantiqini statik metodlarda joylashtirish kodning qayta ishlatilishini va testlashni osonlashtiradi.

Sinf metodlari va statik metodlar o'rtasidagi farqni tushunish muhimdir. Sinf metodlari sinfning o'zi bilan ishlaydi va sinf o'zgaruvchilariga kirish huquqiga ega. Statik metodlar esa to'liq mustaqil funksiyalar bo'lib, faqat mantiqiy jihatdan sinf bilan bog'langan.

Avtomobil sinfidan voris sinflar yaratish turli turdagi avtomobillarni modellashtirish imkonini beradi. Masalan, ElectricCar, HybridCar, SportsCar kabi maxsus sinflari avtomobil bazaviy sinfidan meros olishi mumkin. Har bir voris sinf o'zining maxsus xususiyatlari va metodlariga ega bo'ladi.

Elektr avtomobil sinfi batareya bilan bog'liq qo'shimcha atributlar va metodlarga ega bo'ladi. Batareyani zaryadlash, zaryad darajasini tekshirish, elektr dvigatelni boshqarish kabi maxsus metodlar qo'shiladi. Ota sinfdan meros olingan metodlar override qilinishi yoki kengaytirilishi mumkin.

Gibrid avtomobil sinfi ikkala dvigatel turini boshqarish metodlariga ega bo'ladi. Benzin dvigatelidan elektr dvigateliga o'tish, energiyani qayta tiklash, optimal ish rejimini tanlash kabi murakkab mantiq amalga oshiriladi. Super funksiyasi yordamida ota sinfning metodlari chaqiriladi va kengaytiriladi.

Polimorfizm turli turdagi avtomobillarni bir xil interfeys orqali boshqarish imkonini beradi. Barcha avtomobil turlari bir xil asosiy metodlarga ega bo'lsa ham, har biri o'z implementatsiyasiga ega. Masalan, barcha avtomobillarda tezlikni oshirish metodi mavjud, lekin sport avtomobilida bu metod boshqacha ishlaydi.

Abstrakt bazaviy sinf yaratish umumiy interfeys va majburiy metodlarni belgilash uchun foydalidir. ABC moduli yordamida abstrakt bazaviy sinf yaratiladi va barcha voris sinflar ma'lum metodlarni amalga oshirishi shart qilinadi. Bu sinflar ierarxiyasida izchillikni ta'minlaydi.

Xulosa

Ushbu tadqiqot Python dasturlash tilidagi ob'ektga yo'naltirilgan dasturlashning asosiy tamoyillari va Avtomobil sinfi misolida sinf metodlarini loyihalashning amaliy jihatlarini chuqur tahlil qildi.

Tadqiqot natijalari shuni ko'rsatadiki, sinf metodlarini to'g'ri tanlash va qo'llash dasturiy ta'minotning sifatiga hal qiluvchi ta'sir ko'rsatadi. Ekzemplar metodlari (masalan, tezlikni_oshirish) avtomobilning shaxsiy holatini (tezlik, dvigatel holati)

boshqarishda asosiy rol o'ynaydi. Sinf metodlari (classmethod), odatda, alternativ konstruktorlar (from_vin_number) yoki sinf darajasidagi statistik ma'lumotlarni boshqarish uchun foydalidir. Statik metodlar (staticmethod) esa mustaqil yordamchi funksiyalar (masalan, yonilg_i_sarfini_hisoblash) uchun ishlatilishi kerak.

Sinf metodlarini loyihalashda inkapsulyatsiya tamoyiliga rioya qilish (masalan, private atributlar va ularga property orqali kirishni ta'minlash) avtomobil ob'ekting ichki barqarorligini kafolatlaydi. Meros va polimorfizm esa murakkab avtomobil ierarxiyalarini (Masalan, ElectricCar) yaratishda kodning qayta ishlatilishini va moslashuvchanligini ta'minlaydi.

Xulosa qilib aytganda, ishonchli, miqyoslanuvchan va qo'llab-quvvatlanadigan dasturiy ta'minotni yaratish ob'ektga yo'naltirilgan tamoyillarga qat'iy rioya qilishni, PEP 8 standartlari bo'yicha sifatli kodlashni va metodlarni to'g'ri hujjatlashtirish hamda testlashni talab qiladi. Avtomobil sinfi kabi real hayotiy misollar bu nazariy tamoyillarni amaliyotga samarali tatbiq etish imkonini beradi.

Foydalanilgan Adabiyotlar

1. Van Rossum G, Drake FL. Python Tutorial. Python darsligi. Sentr for National Research Initiatives.
2. Gamma E, Helm R, Johnson R, Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Dizayn Patternlari: Qayta Ishlatiladigan Ob'ektga Yo'naltirilgan Dasturiy Ta'minot Elementlari. Addison-Wesley nashriyoti.
3. Liskov B, Guttag J. Program Development in Java: Abstraction, Specification, and Object-Oriented Design. Javada Dastur Yaratish: Abstraksiya, Spetsifikatsiya va Ob'ektga Yo'naltirilgan Dizayn. Addison-Wesley nashriyoti.
4. Martin RC. Agile Software Development, Principles, Patterns, and Practices. Epchil Dasturiy Ta'minotni Ishlab Chiqish, Tamoyillar, Patternlar va Amaliyotlar. Prentice Hall nashriyoti.
5. Peeters M. Python: The Method, Class, Static Demystified. Python: Metod, Sinf, Statikning Sirli Jihatlari. Real Python.
6. Python Software Foundation. PEP 8 – Style Guide for Python Code. Python Kodu Uchun Stil Bo'yicha Qo'llanma. Python Rasmiy Hujjatlari.
7. Zelle JM. Python Programming: An Introduction to Computer Science. Python Dasturlash: Kompyuter Faniga Kirish. Franklin, Beedle & Associates nashriyoti.