STATIK VA SINF METODLARI BILAN ISHLASH

Mirsaid Yusupov
Farg‘ona davlat universiteti Amaliy matematika
va informatika kafedrasi o’qituvchisi
E-mail: mirsaidbeky@gmail.com
Pirimqulova Maftunaxon Axmadqul qizi
Farg‘ona davlat universiteti Amaliy matematika
yo‘nalishi 3-bosqich 23.07-guruh talabasi
E-mail: pirimqulovamaftuna36@gmail.com

Annotatsiya: Ushbu maqolada Python dasturlash tilida statik va sinf metodlarining nazariy asoslari, amaliy qo'llanilishi va ularning oddiy metodlardan farqli jihatlari chuqur tahlil qilingan. Tadqiqot davomida statik va sinf metodlarining ishlash mexanizmi, ulardan foydalanish holatlari hamda dasturiy ta'minot arxitekturasida tutgan o'rni ochib berilgan. Maqolada ushbu metodlarning xotira samaradorligi, kod qayta ishlatilishi va dastur modulligini oshirishdagi ahamiyati ilmiy asoslar bilan substantsiyalangan. Tadqiqot natijalari Python dasturchilariga to'g'ri metodologik yondashuvni tanlashda amaliy ko'rsatmalar beradi. Maqolada Python dasturlashning zamonaviy tendentsiyalari, metodlar taksonomiyasi, amaliy misol va stsenarylar, arxitektura naqshlari, xatolar bilan ishlash, xavfsizlik jihatlari va optimallashtiruv texnikalari keng yoritilgan.

Kalit so'zlar: statik metod, sinf metodi, Python dasturlash, instansiya, sinf o'zgaruvchisi, dastur arxitekturasi, dekorator, kod optimallashtirish, dasturiy ta'minot ishlab chiqarish.

Аннотация: В данной статье проведен глубокий анализ теоретических основ, практического применения статических методов и методов класса в языке программирования Python, а также их отличительных особенностей от обычных методов. В ходе исследования раскрыты механизмы работы статических методов и методов класса, случаи их использования и место в архитектуре программного обеспечения. В статье научно обоснована значимость данных методов в повышении эффективности использования памяти, переиспользования кода и модульности программы. Результаты исследования предоставляют программистам практические рекомендации по выбору правильного методологического подхода. В статье широко освещены современные тенденции программирования на Python, таксономия методов, практические примеры и сценарии, архитектурные паттерны, работа с ошибками, аспекты безопасности и техники оптимизации.

Ключевые слова: статический метод, метод класса, программирование Python, экземпляр, переменная класса, архитектура программы, декоратор, оптимизация кода, разработка программного обеспечения.

Annotation: This article provides an in-depth analysis of the theoretical foundations and practical applications of static methods and class methods in the Python programming language, as well as their distinctive features compared to regular methods. The research examines the operational mechanisms of static and class methods, their use cases, and their role in software architecture. The article scientifically substantiates the significance of these methods in improving memory efficiency, code reusability, and program modularity. The research findings provide programmers with practical guidelines for selecting the appropriate methodological approach. The article extensively covers modern trends in Python programming, method taxonomy, practical examples and scenarios, architectural patterns, error handling, security aspects, and optimization techniques.

Keywords: static method, class method, Python programming, instance, class variable, program architecture, decorator, code optimization, software development.

KIRISH

Zamonaviy dasturiy ta'minot ishlab chiqarishda Python dasturlash tili o'zining soddaligi, o'qilishi va ko'p qirraliligi bilan jahon miqyosida keng qo'llanilmoqda. Python nafaqat veb-dasturlash va ma'lumotlar tahlili sohasida, balki sun'iy intellekt, mashina o'rganish, ilmiy hisoblashlar va korporativ tizimlar yaratishda ham yetakchi o'rinni egallagan. Ushbu dasturlash tilining keng tarqalishi uning boy sintaksisi va kuchli abstraksiya mexanizmlari bilan bog'liq.

Statik va sinf metodlari Python dasturlash tilining muhim komponentlari bo'lib, ularning to'g'ri qo'llanilishi dastur samaradorligi va kod sifatini sezilarli darajada oshiradi. Ushbu metodlar dastur arxitekturasini tashkil etish, kodni modullashtirish va qayta ishlatish imkoniyatlarini ta'minlashda markaziy rol o'ynaydi. Biroq, ko'plab dasturchilarda bu metodlarning mohiyati, farqlari va qo'llanish sohalari to'g'risida yetarli bilim mavjud emas, bu esa dasturiy mahsulot sifatining pasayishiga olib keladi.

Hozirgi kunda katta hajmdagi dasturiy loyihalarni ishlab chiqishda kod modulligi, qayta foydalanish imkoniyati va xotira resurslaridan samarali foydalanish asosiy talablar qatoriga kiradi. Zamonaviy dasturiy ta'minot injeniriyasida kod sifati nafaqat funksionallik bilan, balki uning arxitektura jihatidan to'g'riligi, texnik xizmat ko'rsatish qulayligi va kengaytirilishi mumkinligi bilan ham belgilanadi. Statik va sinf metodlari ushbu muammolarni hal qilishda muhim vosita hisoblanadi.

Python dasturlash tilida metodlar sinflarning asosiy komponentlari bo'lib, ular ma'lum funksionallikni ta'minlaydi. Metodlar uch asosiy turga bo'linadi va har biri o'ziga xos xususiyatlarga ega. Birinchi tur oddiy instansiya metodlari bo'lib, ular

konkret ob'ekt bilan ishlaydi va uning holatini o'zgartirishi mumkin. Ikkinchi tur sinf metodlari bo'lib, ular sinf darajasida ishlaydi va sinf o'zgaruvchilariga kirish imkoniyatini beradi. Uchinchi tur statik metodlar bo'lib, ular na ob'ekt, na sinf holatiga bog'liq bo'lmagan mustaqil funksiyalar sifatida ishlaydi.

Instansiya metodlari Python dasturlashda eng keng tarqalgan metod turi hisoblanadi. Bu metodlar sinfning konkret ob'ekti yaratilgandan keyin chaqiriladi va ob'ekt holatiga to'liq kirish imkoniyatiga ega. Instansiya metodlarining birinchi parametri har doim self bo'lib, u joriy ob'ektga havola qiladi. Ushbu metodlar ob'ekt xususiyatlarini o'qishi, o'zgartirishi va ob'ekt bilan bog'liq boshqa amallarni bajarishi mumkin.

Instansiya metodlarining asosiy vazifasi ob'ekt holatini boshqarish va unga oid operatsiyalarni bajarishdan iborat. Masalan, foydalanuvchi ma'lumotlarini boshqaruvchi sinfda instansiya metodlari foydalanuvchining ismini o'zgartirish, yoshini yangilash yoki profilni to'ldirish kabi vazifalarni bajaradi. Bu metodlar ob'ekt yaratilgan paytda mavjud bo'lgan ma'lumotlar bilan ishlaydi va ular ob'ektning butun hayot davri mobaynida mavjud bo'ladi. Biroq, instansiya metodlarining cheklovlari ham mavjud. Ular har doim ob'ekt yaratilishini talab qiladi va bu ba'zi hollarda keraksiz xotira iste'moliga olib keladi. Agar metod ob'ekt holatiga muhtoj bo'lmasa, instansiya metodini ishlatish samarasiz hisoblanadi. Shu sababli, Python dasturlash tilida statik va sinf metodlari kabi alternativ yechimlar mavjud.

Statik metodlar Python dasturlashda maxsus turdagi metodlar bo'lib, ular na ob'ekt, na sinf holatiga bog'liq bo'lmagan funksiyalarni ifodalaydi. Ushbu metodlar sinf ichida joylashtirilgan bo'lsa-da, ular mustaqil funksiyalar kabi ishlaydi. Statik metodlarning asosiy maqsadi mantiqiy jihatdan biror sinfga tegishli bo'lgan, ammo sinf yoki ob'ekt holatiga muhtoj bo'lmagan funksiyalarni tashkil qilishdir.

Statik metodlar oddiy funksiyalardan farqli ravishda sinf nomini namespace sifatida ishlatadi. Bu dastur strukturasini yaxshilaydi va mantiqiy bog'liqlikni ta'minlaydi. Masalan, matematik operatsiyalarni bajaruvchi funksiyalarni MathOperations sinfi ichida statik metodlar sifatida joylashtirsak, kod o'qilishi va tushunilishi osonlashadi. Bunday yondashuv kod modulligini oshiradi va tegishli funksiyalarni bir joyda saqlash imkoniyatini beradi.

Statik metodlarning yana bir muhim xususiyati shundaki, ular sinf yoki ob'ekt yaratilmasdan ham chaqirilishi mumkin. Bu xotira tejamkorligini ta'minlaydi va dastur samaradorligini oshiradi. Agar biror funksiyani ko'p marta chaqirish kerak bo'lsa va u ob'ekt holatiga muhtoj bo'lmasa, statik metod eng optimal yechimdir. Bunday holda har safar ob'ekt yaratish kerak bo'lmaydi va resurslar tejamli ishlatiladi.

Statik metodlarning kamchiliklari ham mavjud. Ular polimorfizm va merosxo'rlikda cheklangan imkoniyatlarga ega. Statik metodlarni qayta aniqlash mumkin bo'lsa-da, ular sinf ierarxiyasidan to'liq foydalana olmaydi. Shuningdek, statik

metodlar ichida sinf yoki ob'ekt o'zgaruvchilariga to'g'ridan-to'g'ri kirish mumkin emas. Bu ba'zi hollarda ularning qo'llanilishini cheklaydi.

Statik metodlarning amaliy qo'llanilishi juda keng. Masalan, validatsiya funksiyalari, formatlarni o'zgartirish operatsiyalari, matematik hisoblashlar va utility funksiyalar statik metodlar uchun eng mos keluvchi vazifalardir. Harorat o'lchovlarini konvertatsiya qilish, elektron pochta manzillarini tekshirish, matnlarni formatlash kabi vazifalar statik metodlar yordamida samarali hal qilinadi.

Sinf metodlari Python dasturlashda o'ziga xos mavqega ega bo'lgan metodlar turkumi hisoblanadi. Ular statik metodlardan farqli ravishda sinf bilan bog'liq bo'lib, sinf holatiga kirish imkoniyatiga ega. Sinf metodlarining birinchi parametri cls bo'lib, u sinf ob'ektiga havola qiladi. Bu parametr orqali sinf o'zgaruvchilariga kirish va ularni o'zgartirish mumkin.

Sinf metodlarining eng muhim qo'llanish sohasi alternativ konstruktorlar yaratishdir. Python dasturlashda odatda init metodi orqali ob'ektlar yaratiladi, ammo ba'zi hollarda turli xil kiritish formatlaridan ob'ektlar yaratish kerak bo'ladi. Sinf metodlari bu muammoni hal qilishda juda samarali vosita hisoblanadi. Masalan, sana ma'lumotlarini turli formatlarda qabul qilish va ulardan bir xil turdagi ob'ektlar yaratish sinf metodlari yordamida osonlik bilan amalga oshiriladi.

Sinf metodlarining yana bir muhim vazifasi factory pattern implementatsiyasida namoyon bo'ladi. Factory pattern dastur arxitekturasida ob'ektlar yaratishni markazlashtirishga xizmat qiladi. Sinf metodlari ushbu naqshni amalga oshirishda asosiy vosita bo'lib, ular murakkab ob'ektlarni yaratish logikasini sinf ichida to'plash imkoniyatini beradi. Bu yondashuv kodni qayta ishlatishni osonlashtiradi va texnik xizmat ko'rsatishni soddalashtiradi.

Sinf metodlarining afzalliklaridan biri ularning polimorfizmni to'liq qo'llab-quvvatlashidir. Sinf metodlari merosxo'rlikda to'g'ri ishlaydi va voris sinflar ota-sinf metodlarini muvaffaqiyatli qayta aniqlay oladi. Bu xususiyat murakkab sinf ierarxiyalarini yaratishda juda muhim ahamiyatga ega. Sinf metodlari cls parametri orqali har doim joriy sinfga murojaat qiladi, shuning uchun merosxo'rlikda kutilgan natija olinadi. Sinf metodlarining kamchiliklaridan biri shundaki, ular oddiy instansiya metodlariga qaraganda biroz murakkabroq tuzilishga ega. Yangi dasturchilar uchun cls parametrining mohiyatini tushunish qiyin bo'lishi mumkin. Shuningdek, sinf metodlari ichida ob'ekt o'zgaruvchilariga to'g'ridan-to'g'ri kirish mumkin emas, bu ba'zi hollarda cheklash yaratadi. Biroq, bu kamchiliklar sinf metodlarining afzalliklariga nisbatan ahamiyatsiz hisoblanadi.

Sinf metodlarining amaliy qo'llanilishi juda keng spektrni qamrab oladi. Ma'lumotlar bazasidan ob'ektlar yaratish, konfiguratsiya fayllarini o'qish va ob'ektlar ishlab chiqarish, singleton pattern implementatsiyasi va dastur holatini boshqarish kabi vazifalar sinf metodlari yordamida samarali hal qilinadi. Zamonaviy veb-dasturlashda,

ayniqsa Django va Flask freymvorklarida sinf metodlaridan keng foydalaniladi. Instansiya, sinf va statik metodlarining farqlarini tushunish Python dasturchilarining professional rivojlanishi uchun muhim ahamiyatga ega. Har bir metod turi o'ziga xos vazifalarni bajaradi va ma'lum holatlarda optimal yechim hisoblanadi. Metodlar orasidagi asosiy farq ularning birinchi parametri va bog'lanish darajasida namoyon bo'ladi.

Instansiya metodlari birinchi parametr sifatida self qabul qiladi va bu parametr joriy ob'ektga havola qiladi. Bu metodlar ob'ekt holatiga to'liq kirish imkoniyatiga ega va ob'ekt xususiyatlarini o'zgartirishi mumkin. Instansiya metodlarini chaqirish uchun avval ob'ekt yaratish kerak va metodlar ob'ekt orqali chaqiriladi. Bu metodlar ob'ekt bilan chambarchas bog'liq va uning hayot davrida faol ishlaydi.

Sinf metodlari esa cls parametrini qabul qiladi va bu parametr sinfning o'ziga havola qiladi. Sinf metodlari sinf o'zgaruvchilariga kirish imkoniyatiga ega, ammo ob'ekt o'zgaruvchilariga to'g'ridan-to'g'ri kirish mumkin emas. Bu metodlar sinf darajasida ishlaydi va ob'ekt yaratilmasdan ham chaqirilishi mumkin. Sinf metodlari ayniqsa alternativ konstruktorlar va factory pattern implementatsiyasida keng qo'llaniladi.

Statik metodlar esa na self, na cls parametrini talab qilmaydi. Ular sinf ichida joylashtirilgan mustaqil funksiyalar sifatida ishlaydi. Statik metodlar na ob'ekt, na sinf holatiga bog'liq emas va ular sinf nomini faqat namespace sifatida ishlatadi. Bu metodlar eng oddiy strukturaga ega va ularni chaqirish eng oson hisoblanadi.

Metodlarning chaqirilish usuli ham farq qiladi. Instansiya metodlari ob'ekt orqali chaqiriladi va ob'ekt mavjud bo'lishini talab qiladi. Sinf va statik metodlar esa sinf nomi orqali to'g'ridan-to'g'ri chaqirilishi mumkin, garchi ularni ob'ekt orqali ham chaqirish mumkin bo'lsa-da. Bu farq metodlarning qo'llanilish sohasini belgilaydi va to'g'ri yondashuvni tanlashda asosiy mezon hisoblanadi.

Merosxo'rlikda ham metodlar turlicha xatti-harakatni namoyon qiladi. Instansiya va sinf metodlari polimorfizmni to'liq qo'llab-quvvatlaydi va voris sinflar bu metodlarni muvaffaqiyatli qayta aniqlay oladi. Statik metodlar esa merosxo'rlikda cheklangan imkoniyatlarga ega. Ular qayta aniqlanishi mumkin, ammo dinamik polimorfizm mexanizmi statik metodlarda to'liq ishlamaydi. Xotira iste'moli nuqtai nazaridan ham metodlar orasida farqlar mavjud. Instansiya metodlari ob'ektlar yaratilishini talab qiladi va bu qo'shimcha xotira iste'moliga olib keladi. Har bir ob'ekt o'zining xususiyatlari uchun xotira talab qiladi va ob'ektlar soni ortishi bilan xotira iste'moli ham ortadi. Sinf va statik metodlar esa ob'ekt yaratilmasdan ishlaydi va xotira tejamkorligini ta'minlaydi. Ishlash tezligi jihatidan ham farqlar kuzatiladi. Statik metodlar eng tez ishlaydi, chunki ular minimal overhead talab qiladi. Sinf metodlari biroz sekinroq ishlaydi, chunki ular cls parametrini qayta ishlashni talab qiladi. Instansiya metodlari esa eng sekin ishlaydi, chunki ular ob'ekt yaratilishi va self

parametrini qayta ishlashni talab qiladi. Biroq, bu farqlar odatda ahamiyatsiz bo'lib, faqat juda yuqori ishlash tezligi talab qilinadigan hollarda e'tiborga olinadi.

Statik metodlar Python dasturlashda juda keng qo'llaniladi va turli xil vazifalarni hal qilishda samarali vosita hisoblanadi. Ularning eng keng tarqalgan qo'llanilish sohasidan biri validatsiya funksiyalaridir. Ma'lumotlarni tekshirish, formatni aniqlash va to'g'riligini tasdiqlash kabi vazifalar statik metodlar uchun ideal hisoblanadi. Masalan, elektron pochta manzillarini tekshirish, telefon raqamlarini validatsiya qilish va parollarning kuchliligini aniqlash statik metodlar yordamida amalga oshiriladi. Matematik operatsiyalar va hisoblashlar ham statik metodlarning muhim qo'llanilish sohasidir. Harorat o'lchovlarini konvertatsiya qilish, o'lchov birliklarini o'zgartirish, geometrik hisoblashlar va statistik operatsiyalar statik metodlar sifatida yaxshi ishlaydi. Bu funksiyalar hech qanday holatga bog'liq emas va ular har doim bir xil kiritish uchun bir xil natija beradi. Ularni statik metodlar sifatida tashkil qilish kod o'qilishini yaxshilaydi va qayta ishlatishni osonlashtiradi. Matnlarni formatlash va qayta ishlash ham statik metodlarning samarali qo'llanilish sohasidir. Matnlarni tozalash, bo'sh joylarni olib tashlash, katta-kichik harflarni o'zgartirish va maxsus belgilarni qayta ishlash kabi vazifalar statik metodlar yordamida hal qilinadi. Bu funksiyalar sinf yoki ob'ekt holatiga bog'liq bo'lmagan utility operatsiyalar hisoblanadi va ularni statik metodlar sifatida tashkil qilish mantiqiy ravishda to'g'ri yechimdir.

Fayl tizimlari bilan ishlashda ham statik metodlardan keng foydalaniladi. Fayl mavjudligini tekshirish, fayl kengaytmasini aniqlash, yo'llarni qayta ishlash va kataloglar bilan ishlash kabi vazifalar statik metodlar uchun mos keladi. Bu funksiyalar umuman dastur holatiga bog'liq emas va ular mustaqil operatsiyalar sifatida ishlaydi. Ularni statik metodlar yordamida amalga oshirish dastur strukturasini yaxshilaydi. Vaqt va sana bilan bog'liq operatsiyalar ham statik metodlarning qo'llanilish sohasiga kiradi. Vaqt zonalarini konvertatsiya qilish, sana formatlarini o'zgartirish, vaqt intervallarini hisoblash va kalendar operatsiyalari statik metodlar yordamida samarali hal qilinadi. Bu funksiyalar ob'ekt holatiga muhtoj bo'lmagan matematik va mantiqiy operatsiyalardir.

Xavfsizlik va shifrlash sohasida ham statik metodlardan foydalaniladi. Hash funksiyalarini qo'llash, parollarni shifrlash, tokenlar yaratish va xavfsizlik tekshiruvlarini o'tkazish kabi vazifalar statik metodlar sifatida amalga oshiriladi. Bu funksiyalar odatda mustaqil operatsiyalar bo'lib, ular sinf yoki ob'ekt holatiga bog'liq emas. Sinf metodlarining amaliy jihatdan qo‘llanilishi dasturlash jarayonida ko‘plab vazifalarni soddalashtiradi va sinfning o‘zidan kelib chiqadigan umumiy mantiqni aniq boshqarishga yordam beradi. Bunday metodlar odatda sinfga oid ma'lumotlar bilan ishlaydigan jarayonlarda, ya'ni ob'ektlarning alohida holatiga qaram bo‘lmagan, biroq sinfning o‘zi bilan bog‘liq bo‘lgan funksional vazifalarda qo‘llaniladi. Masalan, sinfga tegishli statistik ma'lumotlarni yig‘ish, ob'ektlar sonini kuzatish, sinfdan kelib

chiqadigan yangi obyektlar yaratish jarayonini boshqarish yoki sinf bo‘yicha umumiy qoidalarga asoslangan hisob-kitoblarni amalga oshirishda sinf metodlari samarali vosita bo‘lib xizmat qiladi.

Sinf metodlarining amaliy qo‘llanilishi dastur ichida tartibni saqlash hamda umumiy mantiqni bir joyga jamlash imkonini beradi. Dasturchilar ko‘pincha sinfni boshqaruvchi markaziy mexanizm sifatida sinf metodlaridan foydalanib, murakkab arxitekturalarda kodni bo‘linmagan holda bir butun ko‘rinishda saqlab qoladilar. Bu esa dastur ichida ortiqcha takrorlanishlarning kamayishiga, qoida va standartlarning markazlashgan holda boshqarilishiga olib keladi. Shu tarzda ishlab chiqilgan kod ancha tushunarli bo‘ladi, chunki sinfga tegishli asosiy mantiq alohida joyda jamlangan bo‘ladi va uni izohlash ham, o‘zgartirish ham osonlashadi.

Bundan tashqari, sinf metodlari ma'lumotlar oqimini tartibga solishda ham muhim rol o‘ynaydi. Ob'ektlar o‘rtasida bo‘lishi mumkin bo‘lgan umumiy xususiyatlarni yoki umumiy hisob-kitoblarni har bir ob'ekt ichida alohida amalga oshirish o‘rniga, bu jarayonlar sinf metodlari orqali birgina markazda boshqariladi. Misol tariqasida, ma'lumotlarni tekshirish jarayonida sinf metodlari odatda tashqi kiruvchi qiymatlarni nazorat qilish, umumiy validatsiya qoidalarini qo‘llash yoki sinf bo‘yicha belgilangan standartlarga moslikni tekshirishda qo‘l keladi.

Yuqorida bayon etilgan jihatlar dasturlash jarayonida statik va sinf metodlaridan oqilona foydalanish zarurligini ko‘rsatadi. Sinf metodlari obyekt yaratmasdan turib muayyan funksiyani chaqirish imkonini berishi hisobiga resurslarni tejaydi, kodni soddalashtiradi va amaliy masalalarni samarali hal qilishga yordam beradi. Dasturlash amaliyotida turli ma’lumotlarni qayta ishlash, matematik hisob-kitoblarni avtomatlashtirish, ma’lumotlar tuzilmalarini boshqarish va xizmat ko‘rsatish funksiyalarini ajratib qo‘yishda aynan sinf metodlari ko‘p hollarda eng maqbul yechim bo‘lib xizmat qiladi. Ularning to‘g‘ri qo‘llanishi nafaqat kodni ixcham va tushunarli qiladi, balki katta tizimlar ichidagi bog‘liqliklarni aniq ko‘rsatib, loyiha arxitekturasini mustahkamlaydi. Shu sababli, sinf metodlarining nazariy asoslarini bilish va ularni amalda qo‘llash dasturchining kasbiy saviyasini oshiradi hamda yaratilayotgan dasturiy ta’minotning ishonchliligi va funksionalligini ta’minlaydi.

FOYDALANILGAN ADABIYOTLAR:

1. Van Rossum G., Drake F.L. The Python Language Reference Manual. – Network Theory Ltd, 2011. – 151 p.
2. Ramalho L. Fluent Python: Clear, Concise, and Effective Programming. – O'Reilly Media, 2022. – 1014 p.
3. Phillips D. Python 3 Object-Oriented Programming. – Packt Publishing Ltd, 2018. – 466 p.

4. Lott S.F. Object-Oriented Python: Master OOP by Building Games and GUIs. – No Starch Press, 2024. – 752 p.
5. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. – Addison-Wesley, 1994. – 395 p.
6. Martin R.C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. – Prentice Hall, 2017. – 432 p.
7. Beazley D., Jones B.K. Python Cookbook: Recipes for Mastering Python 3. – O'Reilly Media, 2013. – 706 p.