

**CSV KUTUBXONASI. .CSV FAYLLAR BILAN ISHLASH
AFZALLIKLARI****Tojimamatov Israil Nurmatovich***Fargʻona davlat universiteti amaliy matematika va informatika kafedrası**katta oʻqituvchisi*israiltojimatov@gmail.com**Satinova Gulshanoy Toʻlanboy qizi***Fargʻona davlat universiteti talabasi*satinovagulshanoy@gmail.com

Annotatsiya: Zamonaviy axborot texnologiyalari rivojlanib borar ekan, katta hajmdagi maʼlumotlar bilan ishlash usullari dolzarb masalaga aylanmoqda. Ushbu maqolada Python dasturlash tilidagi csv moduli asosida strukturaviy oddiylikka ega .csv (Comma-Separated Values) fayllar bilan ishlashning texnik imkoniyatlari va afzalliklari tahlil qilinadi. Formatning yengil tuzilishi, platformalararo mosligi va maʼlumotlar almashinuvidagi roli ilmiy-nazariy asosda koʻrib chiqiladi. Shu bilan birga, .csv formatining JSON va XML kabi zamonaviy maʼlumotlar formatlari bilan solishtirmasi amaliy misollar orqali tahlil qilinadi.

Kalit soʻzlar: : CSV formati, csv moduli, Python, fayl almashinuvi, strukturaviy maʼlumotlar, platformaga mustaqillik, avtomatlashtirish, maʼlumotlar tahlili.

Annotation: As modern information technologies develop, methods for working with large amounts of data are becoming an urgent issue. This article analyzes the technical capabilities and advantages of working with structurally simple .csv (Comma-Separated Values) files based on the csv module in the Python programming language. The format's lightweight structure, cross-platform compatibility, and role in data exchange are considered on a scientific and theoretical basis. At the same time, a comparison of the .csv format with modern data formats such as JSON and XML is analyzed through practical examples.



Keywords: CSV format, csv module, Python, file exchange, structured data, platform independence, automation, data analysis.

Аннотация: С развитием современных информационных технологий актуальным становится вопрос методов работы с большими объемами данных. В данной статье анализируются технические возможности и преимущества работы с файлами .csv (значения, разделенные запятыми), которые имеют структурную простоту, основанную на модуле csv в языке программирования Python. На научной и теоретической основе рассматриваются легкая структура формата, кроссплатформенная совместимость и роль в обмене данными. При этом на практических примерах анализируется сравнение формата .csv с современными форматами данных, такими как JSON и XML.

Ключевые слова: Формат CSV, модуль csv, Python, обмен файлами, структурированные данные, независимость от платформы, автоматизация, анализ данных.

Kirish. Axborot texnologiyalarining jadal rivojlanishi sharoitida ma'lumotlarni saqlash, uzatish va ularni tahlil qilish muhim ilmiy-texnik masalaga aylanmoqda. Ushbu jarayonlarda ma'lumotlar formatining o'rni beqiyosdir. Ayniqsa, tuzilishi oddiy, o'qilishi tez va platformalararo moslashuvchan formatlar keng qo'llanilmoqda. Bunday formatlardan biri — Comma-Separated Values (CSV), ya'ni vergul bilan ajratilgan qiymatlar formatidir.

CSV format ma'lumotlarni qatordan-qatorga, ustundan-ustunga joylashtirishga asoslangan bo'lib, ularning har biri matnli fayl satri ko'rinishida ifodalanadi. Ushbu maqolada Python dasturlash tilining csv moduli yordamida ushbu format bilan ishlash texnologiyasi, uning afzalliklari va amaliy jihatlari chuqur tahlil qilinadi.

CSV (vergul bilan ajratilgan qiymatlar) formati elektron jadvallar va ma'lumotlar bazalari uchun eng keng tarqalgan import va eksport formatidir. CSV formati formatni standartlashtirilgan tarzda tasvirlashga urinishlardan oldin ko'p yillar davomida ishlatilgan RFC 4180. Aniq belgilangan standartning yo'qligi, ko'pincha



turli ilovalar tomonidan ishlab chiqarilgan va iste'mol qilinadigan ma'lumotlarda nozik farqlar mavjudligini anglatadi. Bu farqlar CSV fayllarini bir nechta manbalardan qayta ishlashni bezovta qilishi mumkin. Shunday bo'lsa-da, chegaralovchilar va tirnoq belgilari turlicha bo'lsa-da, umumiy format etarlicha o'xshash bo'lib, dasturchidan ma'lumotlarni o'qish va yozish tafsilotlarini yashirib, bunday ma'lumotlarni samarali boshqara oladigan yagona modul yozish mumkin.

Modul `csv` jadval ma'lumotlarini CSV formatida o'qish va yozish uchun sinflarni amalga oshiradi. Bu dasturchilarga Excel tomonidan qo'llaniladigan CSV formatining aniq tafsilotlarini bilmasdan, "ushbu ma'lumotlarni Excel tomonidan afzal ko'rgan formatda yozing" yoki "Excel tomonidan yaratilgan ushbu fayldan ma'lumotlarni o'qing" deyish imkonini beradi. Dasturchilar, shuningdek, boshqa ilovalar tomonidan tushuniladigan CSV formatlarini tasvirlashlari yoki o'zlarining maxsus maqsadli CSV formatlarini belgilashlari mumkin. Modul va ob'ektlar `csv`-ketlikni o'qiydi va yozadi. Dasturchilar, shuningdek, va sinflari yordamida ma'lumotlarni lug'at shaklida o'qishlari va yozishlari mumkin `.readerwriter.DictReaderDictWriter`. Modul `csv` quyidagi funktsiyalarni belgilaydi: `csv`. **o'quvchi** (`csvfile` , dialekt = 'excel' , `**fmtparams`)

Berilgan *csvfayl*dan *satrlarni qayta ishlovchi* o'quvchi ob'ektini qaytaring . `Csvfayl` har biri o'quvchi tomonidan belgilangan `csv` formatida bo'lgan satrlardan iborat bo'lishi kerak. `Csvfayl` ko'pincha faylga o'xshash ob'ekt yoki ro'yxatdir. Agar *csvfile* fayl ob'ekti bo'lsa, u bilan ochilishi kerak . [1] Muayyan CSV dialektiga xos parametrlar to'plamini aniqlash uchun foydalaniladigan ixtiyoriy *dialekt* parametri berilishi mumkin. Bu sinfning pastki sinfining namunasi yoki funktsiya tomonidan qaytarilgan satrlardan biri bo'lishi mumkin . Boshqa ixtiyoriy *fmtparams* kalit so'z argumentlari joriy dialektdagi individual formatlash parametrlarini bekor qilish uchun berilishi mumkin. Dialekt va formatlash parametrlari haqida to'liq ma'lumot olish uchun "Dialektlar va formatlash parametrlari" bo'limiga qarang `.newline="Dialectlist_dialects()`

`Csv` faylidan o'qilgan har bir satr satrlar ro'yxati sifatida qaytariladi. Format parametri belgilanmagan bo'lsa, ma'lumotlar turini avtomatik o'zgartirish amalga



oshirilmaydi QUOTE_NONNUMERIC(bu holda tirnoqsiz maydonlar floatga aylantiriladi).

Qisqacha foydalanish misoli:

```
import csv
```

```
with open('eggs.csv', newline='') as csvfile:
```

```
    spamreader = csv.reader(csvfile, delimiter=' ', quotechar='|')
```

```
    for row in spamreader:
```

```
        print(' '.join(row))
```

```
Spam, Spam, Spam, Spam, Spam, Baked Beans
```

```
Spam, Lovely Spam, Wonderful Spam
```

```
csv. yozuvchi ( csvfile , dialekt = 'excel' , ** fmtparams )
```

Berilgan faylga o'xshash ob'ektda foydalanuvchi ma'lumotlarini ajratilgan satrlarga aylantirish uchun mas'ul bo'lgan yozuvchi ob'ektini qaytaring. *csvfile* usuli bilan har qanday ob'ekt bo'lishi mumkin `write()`. Agar *csvfile* fayl ob'ekti bo'lsa, uni `newline=""` bilan ochish kerak. Muayyan CSV lahjasiga xos parametrlar to'plamini aniqlash uchun ishlatiladigan ixtiyoriy *dialekt* parametri berilishi mumkin. Bu sinfning pastki sinfining namunasi *Dialect*yoki funktsiya tomonidan qaytarilgan satrlardan biri bo'lishi mumkin `list_dialects()`. Boshqa ixtiyoriy *fmtparams* kalit so'z argumentlari joriy dialektdagi individual formatlash parametrlarini bekor qilish uchun berilishi mumkin. Dialektlar va formatlash parametrlari haqida to'liq ma'lumot olish uchun "Dialektlar va formatlash parametrlari" bo'limiga qarang. DB API-ni amalga oshiradigan modullar bilan interfeysni iloji boricha osonlashtirish uchun qiymat `None` bo'sh qator sifatida yoziladi. Bu qayta tiklanadigan transformatsiya bo'lmasa-da, qo'ng'iroqdan qaytarilgan ma'lumotlarni oldindan qayta ishlamasdan, SQL `NULL` ma'lumotlar qiymatlarini CSV fayllariga tashlashni osonlashtiradi `cursor.fetch*`. Boshqa barcha string bo'lmagan ma'lumotlar `str()` yozilgunga qadar ular bilan bog'langan.

Qisqacha foydalanish misoli:

```
import csv
```

```
with open('eggs.csv', 'w', newline='') as csvfile:
```

```
spamwriter = csv.writer(csvfile, delimiter=' ',  
                        quotechar='|', quoting=csv.QUOTE_MINIMAL)  
spamwriter.writerow(['Spam'] * 5 + ['Baked Beans'])  
spamwriter.writerow(['Spam', 'Lovely Spam', 'Wonderful Spam'])
```

```
csv. registr_dialekt ( ism [ , dialekt [ , **fmtparams ] ] )
```

*Dialekt*ni *ism* bilan bog'lang . *nomi* qator bo'lishi kerak. *Dialekt*ni pastki sinfni *Dialect*yoki *fmtparams* kalit so'z argumentlarini yoki ikkalasini ham, kalit so'z argumentlari *dialekt* parametrlarini bekor qilish orqali belgilash mumkin. Lahjalar va formatlash parametrlari haqida to'liq ma'lumot olish uchun "Dialektlar va formatlash parametrlari" bo'limiga qarang .

```
csv. register_dialekt ( nomi )
```

Lahjalar ro'yxatidan *nom* bilan bog'langan *dialekt*ni o'chiring . Agar *ism* ro'yxatdan o'tgan *dialekt* *nomi* bo'lmasa, an *Error*ko'tariladi .

```
csv. get_dialekt ( ism )
```

Ism bilan bog'langan *dialekt*ni qaytaring . Agar *ism* ro'yxatdan o'tgan *dialekt* *nomi* bo'lmasa, an *Error*ko'tariladi . Bu funktsiya o'zgarishni qaytaradi .*Dialect*

```
csv. ro'yxat_dialektlari ( )
```

Barcha ro'yxatga olingan *dialekt*larning nomlarini qaytaring.

```
csv. maydon_size_chegarasi ( [ yangi_chegarasi ] )
```

Tahlil qiluvchi tomonidan ruxsat etilgan joriy maksimal maydon hajmini qaytaradi. Agar *new_limit* berilgan bo'lsa, bu yangi chegaraga aylanadi.

Modul *csv* quyidagi sinflarni belgilaydi:

```
sinf csv. DictReader ( f , maydon
```

```
nomlari = Yo'q , restkey = Yo'q , restval = Yo'q , dialekt = 'excel' , * args , ** kwds )
```

Oddiy o'quvchi kabi ishlaydigan, lekin har bir satrdagi ma'lumotni kalitlari ixtiyoriy maydon nomlaridict parametri bilan berilgan ob'ektga moslashtiradigan ob'ekt yarating .

Maydon nomlari parametri ketma -ketlikdir . Agar *maydon nomlari* o'tkazib yuborilsa, *f* faylining birinchi qatoridagi qiymatlar maydon nomlari sifatida ishlatiladi va natijalardan olib tashlanadi. Agar *maydon nomlari* ko'rsatilgan bo'lsa,



ular ishlatiladi va birinchi qator natijalarga kiritiladi. Maydon nomlari qanday aniqlanganidan qat'i nazar, lug'at ularning asl tartibini saqlaydi.

Agar satrda maydon nomlaridan ko'proq maydonlar bo'lsa, qolgan ma'lumotlar ro'yxatga kiritiladi va *restart tugmasi* bilan belgilangan maydon nomi bilan saqlanadi (birlamchi qiymat None). Agar bo'sh bo'lmagan qatorda maydon nomlaridan kamroq maydonlar bo'lsa, etishmayotgan qiymatlar restval qiymati bilan to'ldiriladi (birlamchi qiymat None).

Boshqa barcha ixtiyoriy yoki kalit so'z argumentlari asosiy readermisolga uzatiladi.

Agar maydon nomlariga uzatilgan argument iterator bo'lsa, u ga majburlanadi list.

3.6-versiyasida o'zgartirildi: Qaytarilgan qatorlar endi turdagi OrderedDict.

3.8 versiyasida o'zgartirildi: Qaytarilgan qatorlar endi dict.

Qisqacha foydalanish misoli:

```
import csv
```

```
with open('names.csv', newline='') as csvfile:
```

```
    reader = csv.DictReader(csvfile)
```

```
    for row in reader:
```

```
        print(row['first_name'], row['last_name'])
```

```
Eric Idle
```

```
John Cleese
```

```
>>> print(row)
```

```
{'first_name': 'John', 'last_name': 'Cleese'}
```

```
sinf csv. DictWriter ( f , maydon
```

```
nomlari , restval = " , ekstrasatsiya = 'ko'tarish' , dialekt = 'excel' , * args , ** kwds )
```

Oddiy yozuvchi kabi ishlaydigan, lekin lug'atlarni chiqish qatorlariga joylashtiradigan ob'ekt yarating. Maydon *nomlarisequence* parametri lug'atdagi qiymatlarni *fwriterow()* fayliga yozish tartibini aniqlaydigan kalitlardan iborat . Ixtiyoriy *restval* parametri, agar lug'atda maydon nomlarida kalit yo'q bo'lsa, yoziladigan qiymatni belgilaydi . Agar usulga o'tkazilgan



lug'atda *maydon* nomlarida topilmagan kalit bo'lsa, ixtiyoriy *ekstrasasiya* parametri qanday harakat qilish kerakligini ko'rsatadi. Agar u ga o'rnatilgan bo'lsa, standart qiymat a ko'tariladi. Agar u ga o'rnatilgan bo'lsa, lug'atdagi qo'shimcha qiymatlar e'tiborga olinmaydi. Har qanday boshqa ixtiyoriy yoki kalit so'z argumentlari asosiy misolga uzatiladi `.writerow()` `'raise'ValueError'ignore'` `writer`. E'tibor bering, sinfdan farqli o'laroq `DictReader`, sinfning `fieldnamesDictWriter` parametri ixtiyoriy emas.

Agar *maydon* nomlariga uzatilgan argument iterator bo'lsa, u ga majburlanadi list. Qisqacha foydalanish misoli:

```
import csv

with open('names.csv', 'w', newline='') as csvfile:
    fieldnames = ['first_name', 'last_name']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerow({'first_name': 'Baked', 'last_name': 'Beans'})
    writer.writerow({'first_name': 'Lovely', 'last_name': 'Spam'})
    writer.writerow({'first_name': 'Wonderful', 'last_name': 'Spam'})
```

Sinf `csv`. **Dialect** konteyner sinfi bo'lib, atributlari qo'sh tirnoq, bo'sh joy, chegaralovchilar va hokazolarni qanday ishlatish bo'yicha ma'lumotlarni o'z ichiga oladi. CSV qat'iy spetsifikatsiyasi yo'qligi sababli, turli ilovalar bir-biridan farq qiluvchi CSV ma'lumotlarini ishlab chiqaradi. `Dialect` misollar qanday `readerv` `writer` misollar harakat qilishini belgilaydi.

Mavjud barcha `Dialect` nomlar tomonidan qaytariladi `list_dialects()` va ular quyidagi kabi ishga tushirish () funksiyalari orqali maxsus `readerv` sinflar bilan ro'yxatdan o'tkazilishi mumkin: `writer__init__`

```
import csv

with open('students.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile, dialect='unix')
```

sinf `csv`. **Excel** Excelda yaratilgan CSV faylining odatiy xususiyatlarini belgilaydi. U dialekt nomi bilan ro'yxatga olingan 'excel'.



Sinf csv. **excel_tab** Excelda yaratilgan TAB bilan ajratilgan faylning odatiy xususiyatlarini belgilaydi. U dialekt nomi bilan ro'yxatga olingan 'excel-tab'.

Sinf csv. **unix_dialekt** UNIX tizimlarida yaratilgan CSV faylining odatiy xususiyatlarini belgilaydi, ya'ni '\n'chiziqli terminatori sifatida foydalanish va barcha maydonlarni iqtibos qilish. U dialekt nomi bilan ro'yxatga olingan 'unix'.

3.2 versiyasida qo'shilgan.

sinf csv. **Sniffer**

Sinf SnifferCSV fayl formatini chiqarish uchun ishlatiladi.

Sinf Snifferikkita usulni taqdim etadi:

hidlash (namuna , chegaralovchilar = Yo'q)

Berilgan *namunaniDialect* tahlil qiling va topilgan parametrlarni aks ettiruvchi kichik sinfni qaytaring . Agar ixtiyoriy *chegaralovchilar* parametri berilgan bo'lsa, u mumkin bo'lgan haqiqiy chegaralovchi belgilarni o'z ichiga olgan qator sifatida talqin qilinadi.

has_header (namuna)

Namuna matnni tahlil qiling (CSV formatida bo'lishi taxmin qilinadi) va Trueagar birinchi qator ustun sarlavhalari qatori bo'lsa, uni qaytaring. Har bir ustunni tekshirganda, namunada sarlavha mavjud yoki yo'qligini baholash uchun ikkita asosiy mezonlardan biri ko'rib chiqiladi: ikkinchidan n-chi qatorlar raqamli qiymatlarni o'z ichiga oladi.

ikkinchidan n-chi qatorlar satrlarni o'z ichiga oladi, bunda kamida bitta qiymat uzunligi ushbu ustunning taxminiy sarlavhasi uzunligidan farq qiladi.

Birinchi qatordan keyin yigirma qator namuna olinadi; agar ustunlar + satrlarning yarmidan ko'pi mezonlarga javob bersa, Trueqaytariladi.

Python dasturlash tilining csv moduli — standart kutubxona bo'lib, foydalanuvchiga CSV fayllarni o'qish va yozishda yuqori darajadagi funksional qulayliklarni taqdim etadi. Modulning tuzilmasi modulli arxitektura prinsipi asosida tashkil etilgan bo'lib, quyidagi asosiy komponentlarni o'z ichiga oladi:



csv.reader - ushbu funksional birlik CSV faylni satrlar bo'yicha o'qib, har bir satrni Python ro'yxati (list) ko'rinishida qaytaradi. Bu modul elementar tahlil, ma'lumotlarni filtratsiya va indekslashda qo'llaniladi.

csv.writer - mazkur komponent CSV faylga yozish operatsiyalarini bajaradi. Yozilayotgan ma'lumotlar Python ro'yxatlari yoki massivlar ko'rinishida bo'ladi. Strukturaviy aniqlikni ta'minlashda ushbu funksiyaning roli katta.

csv.DictReader va **csv.DictWriter** - ushbu metodlar har bir satrni kalit-qiymat (key-value) juftliklaridan iborat dict tipidagi obyekt sifatida ifodalaydi. Bu yondashuv ma'lumotlarning semantik tushunarlilikini oshiradi va kodning o'qiluvchanligini yaxshilaydi.

Modul quyidagi muhim sozlamalarni qo'llab-quvvatlaydi:

- **delimiter=','** — ustunlar orasidagi ajratkich belgisi;
- **quotechar='\"'** — matn qiymatlarini belgilovchi belgi;
- **lineterminator='\\n'** — satr yakunlovchi belgi;
- **encoding='utf-8'** — kodlash formatlari.

Bu parametrlar turli standartlardagi CSV fayllar bilan ishlashda yuqori darajadagi moslashuvchanlikni ta'minlaydi.

Strukturaviy soddalik - CSV format faqatgina matndan iborat bo'lib, uning strukturasi satrlar va ustunlar ko'rinishida aniqlanadi. Bu oddiylik uni har qanday matn muharriri orqali ko'rish va tahrirlash imkonini yaratadi.

Platformadan mustaqillik - CSV fayllari Windows, Linux va macOS kabi operatsion tizimlar orasida to'liq mos keladi. Shuningdek, Excel, Google Sheets, pandas kabi ko'plab dasturiy vositalar bilan to'g'ridan-to'g'ri integratsiyalashadi.

Hajm jihatidan tejamkorlik - JSON va XML formatlaridan farqli o'laroq, CSV fayllar ortiqcha strukturaviy elementlarga ega emas. Bu esa fayl hajmini kamaytiradi, tarmoq orqali tezroq uzatilishini va kam resurs talab qilinishini ta'minlaydi.

Yuqori interoperabilite - CSV formati ko'plab statistik, tahliliy va ma'lumotlar bazasi tizimlari tomonidan qo'llab-quvvatlanadi (masalan:



MySQL, PostgreSQL, R, SPSS). Bu esa uni global ma'lumotlar almashinuvida universal vosita sifatida qo'llash imkonini beradi.

Avtomatlashtirish qulayligi - python csv moduli yordamida ma'lumotlarni avtomatik ravishda o'qish, yozish, filtratsiya qilish va tahlil qilish mumkin. Bu esa ishlab chiqilayotgan dasturlarda samaradorlikni oshiradi.

CSV faylni o'qish:

```
import csv  
with open('xodimlar.csv', newline="", encoding='utf-8') as file:  
    reader = csv.DictReader(file)  
    for row in reader:  
        print(f"{row['Ism']} - {row['Lavozim']}")
```

CSV faylga yozish:

```
import csv  
mahsulotlar = [  
    {'Nomi': 'Noutbuk', 'Narxi': 750},  
    {'Nomi': 'Monitor', 'Narxi': 250}  
]  
with open('mahsulotlar.csv', 'w', newline="", encoding='utf-8') as file:  
    fieldnames = ['Nomi', 'Narxi']  
    writer = csv.DictWriter(file, fieldnames=fieldnames)  
    writer.writeheader()  
    for mahsulot in mahsulotlar:  
        writer.writerow(mahsulot)
```

Natija. O'rganilgan holatlar asosida quyidagi ilmiy asoslangan xulosalarga kelish mumkin:

- CSV fayl formati yengil va samarali tuzilishga ega bo'lib, u zamonaviy dasturlash tizimlarida keng qo'llaniladi.
- Python csv kutubxonasi yordamida ma'lumotlar bilan avtomatik, aniq va tejamkor ishlash mumkin.



- Platformalararo moslashuvchanlik va kam resurs talab etilishi bu formatni yirik tizimlar bilan integratsiyalashuvda maqbul qiladi.

Xulosa. Yuqorida o'rganilgan tahlillar asosida aytish mumkinki, CSV formati — zamonaviy axborot almashinuvi va dasturiy tizimlarda samaradorlikni ta'minlovchi asosiy vositalardan biridir. Uning soddaligi, umumiy standartlarga mosligi, Python platformasidagi yuqori darajadagi qo'llab-quvvatlanishi uni turli sohaga oid amaliy loyihalarda qo'llash imkonini beradi. Ilmiy-tadqiqot loyihalari, tahliliy tizimlar, statistik hisob-kitoblar va dasturiy interfeyslar bu formatdan samarali foydalanmoqda.

FOYDALANILGAN ADABIYOTLAR

1. Python Software Foundation. *Python CSV Documentation*. <https://docs.python.org/3/library/csv.html>
2. McKinney, W. (2018). *Python for Data Analysis*. O'Reilly Media.
3. Wickham, H. (2014). *Tidy Data*. Journal of Statistical Software.
4. Reinsel, D., Gantz, J., & Rydning, J. (2018). *The Digitization of the World From Edge to Core*, IDC White Paper.
5. Open Data Handbook. *Understanding CSV Format*.
6. Wes McKinney. *Data Wrangling with Pandas and CSV*.
7. Tojimamatov, I., & Mirsiddiqova, M. (2025). BERILGANLAR BAZASIDA HAYOTIY SIKL. Модели и методы в современной науке, 4(6), 66-70.
8. Tojimamatov, I., & Siddiqova, G. (2025). TRANZAKSIYALARNI TAQSIMLANGAN TARZDA QAYTA ISHLASH MODELLARI. Современные подходы и новые исследования в современной науке, 4(6), 30-35.
9. Нурмаматович, Т. И., & Рахила, А. (2025). НА ОСНОВЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ПОВЫШЕНИЕ УСТОЙЧИВОСТИ К ПОЛОМКАМ И АВАРИЯМ. YANGI O'ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 2(8), 197-204.
10. Tojimamatov, I., & Ahmataliyeva, S. (2025). BERILGANLARNI MARKAZLASHGAN TARZDA BOSHQARISH



TAMOYILLARI. Академические исследования в современной науке, 4(21), 59-64.

11. Tojimamatov, I., & Adxamova, C. (2025). AMALIY TIZIMLARDA BERILGANLAR BAZASINI BOSHQARISH TIZIMLARI O‘RNI. Академические исследования в современной науке, 4(21), 77-82.

12. Tojimamatov, I., & Fazliddinov, X. (2025). BERILGANLAR BAZASI ADMINISTRATORI VA UNING XUSUSYATLAR. Академические исследования в современной науке, 4(21), 90-95.

13. Karimberdiyevich, O. M., Nurmamatovich, T. I., & Abdulaziz o‘g‘li, Y. M. (2024). BIG DATA SOHASIDAGI XALQARO LOYIHALAR. IZLANUVCHI, 1(1), 39-45.

14. Karimberdiyevich, O. M., Abdulaziz o‘g‘li, Y. M., & Zarifjon o‘g‘li, X. N. (2024). DATA MINING METODLARI VA BOSQICHLARI. YANGI O‘ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(4), 303-311.

15. Nurmamatovich, T. I. (2024). BERILGANLARNING TARMOQ MODELLARI: ODDIY VA MURAKKAB TARMOQ TUZILISHLARI.