



**FARG'ONA DAVLAT UNIVERSITETI
HUZURIDAGI PEDAGOG KADRLARNI
QAYTA TAYYORLASH VA ULARNING
MALAKASINI OSHIRISH MINTAQAVIY
MARKAZI**



**FIZIK JARAYONLARNI
KOMPYUTERDA MODELLASHTIRISH**



Modulning ishchi dasturi Oliy ta'lim, fan va innovatsiyalar vazirligining 2024-yil 27 dekabrda 485-sonli buyrug'i bilan tasdiqlangan oliy ta'lim muassasalari rahbar kadrlarini qayta tayyorlash va malaka oshirish yo'nalishlari o'quv reja va dasturlariga muvofiq ishlab chiqilgan.

Tuzuvchi:

D.Yusupova – FarDU , f.m.f.n.,
Fizika kafedrasida dotsenti.

Taqrizchi:

K.E.Onarqulov- FarDU Fizika
kafedrasida mudiri, f.m.f.d., professor

I. ISHCHI DASTUR

Kirish

Ushbu dastur O‘zbekiston Respublikasining 2020-yil 23-sentabrda tasdiqlangan “Ta’lim to‘g‘risida” Qonuni, O‘zbekiston Respublikasi Prezidentining 2015-yil 12-iyundagi “Oliy ta’lim muassasalarining rahbar va pedagog kadrlarini qayta tayyorlash va malakasini oshirish tizimini yanada takomillashtirish to‘g‘risida” PF-4732-son, 2019-yil 27-avgustdagi “Oliy ta’lim muassasalari rahbar va pedagog kadrlarining uzluksiz malakasini oshirish tizimini joriy etish to‘g‘risida” PF-5789-son, 2019-yil 8-oktabrdagi “O‘zbekiston Respublikasi oliy ta’lim tizimini 2030-yilgacha rivojlantirish konsepsiyasini tasdiqlash to‘g‘risida” PF-5847-son, 2020-yil 29-oktabrdagi “Ilm-fanni 2030-yilgacha rivojlantirish konsepsiyasini tasdiqlash to‘g‘risida” PF-6097-son, 2022-yil 28-yanvardagi “2022-2026 yillarga mo‘ljallangan Yangi O‘zbekistonning taraqqiyot strategiyasi to‘g‘risida” PF-60-son, 2023-yil 25-yanvardagi “Respublika ijro etuvchi hokimiyat organlari faoliyatini samarali yo‘lga qo‘yishga doir birinchi navbatdagi tashkiliy chora-tadbirlar to‘g‘risida” PF-14-son, O‘zbekiston Respublikasi Prezidentining 2023-yil 11-sentabrdagi ““O‘zbekiston — 2030” strategiyasi to‘g‘risida” PF-158-son Farmonlari, shuningdek, O‘zbekiston Respublikasi Prezidentining 2024-yil 21-iyundagi “Aholi va davlat xizmatchilarining korrupsiyaga qarshi kurashish sohasidagi bilimlarini uzluksiz oshirish tizimini joriy qilish chora-tadbirlari to‘g‘risida” PQ-228-son, O‘zbekiston Respublikasi Prezidentining 2021-yil 17-fevraldagi “Sun‘iy intellekt texnologiyalarini jadal joriy etish uchun shart-sharoitlar yaratish chora-tadbirlari to‘g‘risida” PQ-4996-son qarorlari va O‘zbekiston Respublikasi Vazirlar Mahkamasining “Oliy ta’lim muassasalari rahbar va pedagog kadrlarining malakasini oshirish tizimini yanada takomillashtirish bo‘yicha qo‘shimcha chora-tadbirlar to‘g‘risida” 2019-yil 23-sentabrdagi 797-son hamda O‘zbekiston Respublikasi Vazirlar Mahkamasining “Oliy ta’lim tashkilotlari rahbar va pedagog kadrlarini qayta tayyorlash va malakasini oshirish tizimini samarali tashkil qilish chora-tadbirlari to‘g‘risida” 2024-yil 11-iyuldagi 415-son Qarorlarida belgilangan ustuvor vazifalar mazmunidan kelib chiqqan holda tuzilgan bo‘lib, u

oliy ta'lim muassasalari pedagog kadrlarining kasb mahorati hamda innovatsion kompetentligini rivojlantirish, sohaga oid ilg'or xorijiy tajribalar, yangi bilim va malakalarni o'zlashtirish, shuningdek amaliyotga joriy etish ko'nikmalarini takomillashtirishni maqsad qiladi.

Dastur doirasida berilayotgan mavzular ta'lim sohasi bo'yicha pedagog kadrlarni qayta tayyorlash va malakasini oshirish mazmuni, sifati va ularning tayyorgarligiga qo'yiladigan umumiy malaka talablari va o'quv rejalari asosida shakllantirilgan bo'lib, uning mazmuni yangi O'zbekistonning taraqqiyot strategiyasi va jamiyatning ma'naviy asoslarini yoritib berish, oliy ta'limning normativ-huquqiy asoslari bo'yicha ta'lim-tarbiya jarayonlarini tashkil etish, pedagogik faoliyatda raqamli kompetensiyalarni rivojlantirish, ilmiy-innovatsion faoliyat darajasini oshirish, pedagogning kasbiy kompetensiyalarini rivojlantirish, ta'lim sifatini ta'minlashda baholash metodikalaridan samarali foydalanish, fizik jarayonlarni kompyuterda modellashtirish asoslarini o'zlashtirish, astronomiya va astrofizikaning zamonaviy muammolarini o'rganish va yutuqlaridan foydalanish bo'yicha tegishli bilim, ko'nikma, malaka va kompetensiyalarni rivojlantirishga yo'naltirilgan.

Modulning maqsadi va vazifalari

“Fizik jarayonlarni kompyuterda modellashtirish” modulining maqsad va vazifalari:

- pedagog kadrlarning innovatsion yondoshuvlar asosida o'quv-tarbiyaviy jarayonlarni yuksak ilmiy-metodik darajada loyihalashtirish, sohadagi ilg'or tajribalar, zamonaviy bilim va malakalarni o'zlashtirish va amaliyotga joriy etishlari uchun zarur bo'ladigan kasbiy bilim, ko'nikma va malakalarini takomillashtirish, shuningdek ularning ijodiy faolligini rivojlantirishdan iborat

Kursning **vazifalariga** quyidagilar kiradi:

“Fizika va astronomiya” yo'nalishida pedagog kadrlarning kasbiy bilim, ko'nikma, malakalarini takomillashtirish va rivojlantirish;

- o'quv jarayonini tashkil etish va uning sifatini ta'minlash borasidagi ilg'or xorijiy tajribalar, zamonaviy yondashuvlarni o'zlashtirish;

“Fizika va astronomiya” yo‘nalishida qayta tayyorlash va malaka oshirish jarayonlarini fan va ishlab chiqarishdagi innovatsiyalar bilan o‘zaro integratsiyasini ta’minlash.

Modul yakunida tinglovchilarning bilim, ko‘nikma va malakalari hamda kompetensiyalariga qo‘yiladigan talablar:

Qayta tayyorlash va malaka oshirish kursining “Fotonikaning zamonaviy masalalari” o‘quv moduli bo‘yicha tinglovchilar quyidagi yangi bilim, ko‘nikma, malaka hamda kompetensiyalarga ega bo‘lishlari talab etiladi:

Tinglovchi:

- *kompyuterli eksperiment bosqichlarini;*
- *dasturiy ta’minot: Spyder/MatlabWork vositalari va tillarini;*
- *kodlarni tuzish: algoritmlar, o‘zgaruvchanlar, ma’lumotlar turlarini;*
- *dasturlar tuzishning dastlabki qadamlari, chiziqli algoritmlar va oddiy misollarini;*
- *tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollarini bilishi kerak.*

Tinglovchi:

- kompyuterda, modellashtirish uchun operatsion tizim tushunchasini bilish;
- Python redaktorlarini (Spyder, PyCharm, VS Code) o‘rnatish;
- munosabatlar, tanlash va solishtirish operatorlari bilan ishlash;
- sonli integrallash va differensial tenglamalarni sonli yechish usullardan foydalanish;
- teylor qatorlari asosida funksiyalarning qiymatini hisoblash;
- grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish **ko‘nikmalariga** ega bo‘lishi lozim.

Tinglovchi:

- sikllar yordamida summa va ko‘paytmalarni topish;

- Chandrasekar va Opegeymer-Volkov chegaralaridan foydalana olish;
- chala tarmoqli va to‘liq tarmoqli algoritmlarni ahamiyatini ochib berish;
- python tiliga kirish: projektни saqlash va tayyorini davom ettirish, sintaksisining asosiy qoidalarini qo‘llash *malakalariga* ega bo‘lishi zarur.

Tinglovchi:

- mexanik harakatlarni modellashtirish va ularni visuallashtirish usullaridan foydalanish *kompetensiyalariga* ega bo‘lishi lozim.

Modul hajmi

«Fizik jarayonlarni kompyuterda modellashtirish» jami 12 soatni – 8 soat nazariy, 8 soat amaliy hamda 12 soat ko‘chma mashg‘ulotni tashkil etadi.

«FIZIK JARAYONLARNI KOMPYUTERDA MODELLASHTIRISH»

O‘QUV MODULINING MAZMUNI

Model va modellashtirish tushunchasi. Modellar turlari va klassifikatsiyasi. Kompyuterda modellashtirish. Xisoblash fizikasi. Modellashtirish fizikasi. Kompyuterli eksperiment bosqichlari. Kompyuterda, modellashtirish uchun operatsion tizim. Dasturiy ta’minot: Spyder/MatlabWork vositalari va tillari. Ularni kompyuterda o‘rnatish yo‘llari: Anaconda, Python for data science. Python redaktorlarini (Spyder, PyCharm, VS Code) o‘rnatish.

Kodlarni tuzish: algoritmlar, o‘zgaruvchanlar, ma’lumotlar turlari. Python tiliga kirish: projektни saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari. Dasturlar tuzishning dastlabki qadamlari, chiziqli algoritmlar va oddiy misollar.

Tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollari. Chala tarmoqli va to‘liq tarmoqli algoritmlar. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash. Mantiqiy operatsiyalar. Sikllar bilan ishlash: sikllar va siklik operatorlar. Sikllar yordamida summa va ko‘paytmalarni topish. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash.

Massivlar: bir o‘lchamli va ikki o‘lchamli massivlarni shakillantirish, elementlari bilan ishlash. Tasodifiy jarayonlar xaqida tushuncha va tasodifiy sonlar

bilan ishlah. Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.

Ekspariment ma'lumotlarining statistik tahlili. Sonli integrallash usullari va algoritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari. Ular yordamida aniq integrallarni hisoblash. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmlari. Xarakat tenglamasini sonli yechish. Mexanik harakatlarni modellashtirish va ularni visuallashtirish usullari.

Modul bo'yicha soatlar taqsimoti

№	Mavzular	Umumiy soat	Jami aydoriya soati	Jumladan		
				nazariy	amaliy	ko'chma mashg'ulot
1	Nazariy mashg'ulot					
1.1	Model va modellashtirish tushunchasi. Modellar turlari va klassifikatsiyasi. Kompyuterda modellashtirish. Xisoblash fizikasi. Modellashtirish fizikasi. Kompyuterli eksperiment bosqichlari. Kompyuterda, modellashtirish uchun operatsion tizim. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for data science.	2	2	2		
1.2	Kodlarni tuzish: algoritmlar, o'zgaruvchanlar, ma'lumotlar turlari.	2	2	2		

	<p>Python tiliga kirish: proyektни saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari.</p> <p>Tarmoqlanuvchi algoritmlar, if va else shart operatrlari bilan ishlash misollari. Chala tarmoqli va to'liq tarmoqli algoritmlar. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash. Mantiqiy operatsiyalar. Sikllar bilan ishlash: sikllar va siklik operatorlar. Sikllar yordamida summa va ko'paytmalarni topish. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash.</p>					
1.3	<p>Massivlar: bir o'lchamli va ikki o'lchamli massivlarni shakillantirish, elementlari bilan ishlash. Tasodifiy jarayonlar xaqida tushuncha va tasodifiy sonlar bilan ishlash. Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.</p>	2	2	2		
1.4	<p>Eksperiment ma'lumotlarining statistika tahlili. Sonli integrallash usullari va algoritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari.</p>	2	2	2		

	Ular yordamida aniq integrallarni hisoblash. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmlari.					
II.	Amaliy mashg'ulot					
2.1.	Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for data science. Python redaktorlarini (Spyder, PyCharm, VS Code) o'rnatish. Dasturlar tuzishning dastlabki qadamlari, chiziqli algoritmlar va oddiy misollar.	2	2		2	
2.2.	Sikllar bilan ishlash: sikllar va siklik operatorlar. Sikllar yordamida summa va ko'paytmalarni topish. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash. Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va histogrammalarni chizish.	2	2		2	
2.	Eksperiment ma'lumotlarining statistik tahlili. Sonli integrallash usullari va algoritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari.	2	2		2	

	Ular yordamida aniq integrallarni hisoblash.					
2.4.	Oddiy differensial tenglamalarni sonli yechish usullari va algoritmlari. Xarakat tenglamasini sonli yechish. Mexanik harakatlarni modellashtirish va ularni visuallashtirish usullari.	2	2		2	
II.	Ko'chma mashg'ulot					
2.5.	Zamonaviy jihozlar hamda innovatsion texnologiyalarni qo'llab faoliyat yuritayotgan oliy ta'lim muassasasiga olib borish					12
III.	Malakaviy attestatsiya					
	Jami	28	16	8	8	12

Amaliy mashg'ulotlarni tashkil etish bo'yicha ko'rsatma va tavsiyalar

Amaliy mashg'ulotlarda tinglovchilar o'quv modullari doirasidagi ijodiy topshiriqlar, keyslar, o'quv loyihalari, texnologik jarayonlar bilan bog'liq vaziyatli masalalar asosida amaliy ishlarni bajaradilar.

Amaliy mashg'ulotlar zamonaviy ta'lim uslublari va innovatsion texnologiyalarga asoslangan holda o'tkaziladi. Bundan tashqari, mustaqil holda o'quv va ilmiy adabiyotlardan, elektron resurslardan, tarqatma materiallardan foydalanish tavsiya etiladi.

Ko'chma mashg'ulotlarni tashkil etish bo'yicha ko'rsatma va tavsiyalar

Ko'chma mashg'ulotlar zamonaviy jihozlar hamda innovatsion texnologiyalarni qo'llab faoliyat yuritayotgan ishlab chiqarish korxonasi va tashkilotlari, oliy ta'lim muassasalari, iqtisodiyot tarmoqlari, ilmiy-tadqiqot va loyiha-konstruktorlik muassasalarida olib boriladi.

Dasturning axborot-metodik ta'minoti

Modulni o'qitish jarayonida ishlab chiqilgan o'quv-metodik materiallar, tegishli soha bo'yicha ilmiy jurnallar, Internet resurslari, multimedia mahsulotlari va boshqa elektron va qog'oz variantdagi manbaalardan foidalaniadi.

ADABIYOTLAR

I. O'zbekiston Respublikasi Prezidentining asarlari

1. Mirziyoev Sh.M. Byuuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. – T.: “O'zbekiston”, 2017. – 488 b.

2. Mirziyoev Sh.M. Milliy taraqqiyot yo'limizni qat'iyat bilan davom ettirib, yangi bosqichga ko'taramiz. 1-jild. – T.: “O'zbekiston”, 2017. – 592 b.

Mirziyoev Sh.M. Xalqimizning roziligi bizning faoliyatimizga berilgan eng oliy bahodir. 2-jild. T.: “O'zbekiston”, 2018. – 507 b.

4. Mirziyoev Sh.M. Niyati ulug' xalqning ishi ham ulug', hayoti yorug' va kelajagi farovon bo'ladi. 3-jild.– T.: “O'zbekiston”, 2019. – 400 b.

5. Mirziyoev Sh.M. Milliy tiklanishdan – milliy yuksalish sari. 4-jild.– T.: “O'zbekiston”, 2020. – 400 b.

II. Normativ-huquqiy hujjatlar

1. O'zbekiston Respublikasining Konstitutsiyasi. – T.: O'zbekiston, 2024

2. O'zbekiston Respublikasining 2020-yil 23-sentabrda qabul qilingan “Ta'lim to'g'risida”gi Qonuni.

3. O'zbekiston Respublikasining “Korrupsiyaga qarshi kurashish to'g'risida”gi Qonuni.

4. O'zbekiston Respublikasi Prezidentining 2015-yil 12-iyundagi “Oliy ta'lim muassasalarining rahbar va pedagog kadrlarini qayta tayyorlash va malakasini oshirish tizimini yanada takomillashtirish to'g'risida”gi PF-4732-sonli Farmoni.

5. O'zbekiston Respublikasi Prezidentining 2019-yil 27-maydagi “O'zbekiston Respublikasida korrupsiyaga qarshi kurashish tizimini yanada takomillashtirish chora-tadbirlari to'g'risida”gi PF-5729-son Farmoni.

6. O'zbekiston Respublikasi Prezidentining 2019-yil 27-avgustdagi “Oliy ta'lim

muassasalari rahbar va pedagog kadrlarining uzluksiz malakasini oshirish tizimini joriy etish to'g'risida"gi PF-5789-sonli Farmoni.

7. O'zbekiston Respublikasi Vazirlar Mahkamasining 2019-yil 23-sentabrdagi "Oliy ta'lim muassasalari rahbar va pedagog kadrlarining malakasini oshirish tizimini yanada takomillashtirish bo'yicha qo'shimcha chora-tadbirlar to'g'risida"gi 797-sonli Qarori.

8. O'zbekiston Respublikasi Prezidentining 2019-yil 8-oktabrdagi "O'zbekiston Respublikasi oliy ta'lim tizimini 2030-yilgacha rivojlantirish konsepsiyasini tasdiqlash to'g'risida"gi PF-5847- sonli Farmoni.

9. O'zbekiston Respublikasi Prezidentining 2020-yil 29-oktabr "Ilm-fanni 2030 yilgacha rivojlantirish konsepsiyasini tasdiqlash to'g'risida"gi PF-6097-sonli Farmoni.

10. O'zbekiston Respublikasi Prezidentining 2021-yil 17-fevraldagi "Sun'iy intellekt texnologiyalarini jadal joriy etish uchun shart-sharoitlar yaratish chora-tadbirlari to'g'risida"gi PQ-4996-son Qarori.

11. O'zbekiston Respublikasi Prezidentining 2022-yil 28-yanvardagi "2022-2026 yillarga mo'ljallangan Yangi O'zbekistonning taraqqiyot strategiyasi to'g'risida"gi PF-60-son Farmoni.

12. O'zbekiston Respublikasi Prezidentining 2023-yil 25-yanvardagi "Respublika ijro etuvchi hokimiyat organlari faoliyatini samarali yo'lga qo'yishga doir birinchi navbatdagi tashkiliy chora-tadbirlar to'g'risida"gi PF-14-sonli Farmoni.

13. O'zbekiston Respublikasi Prezidentining 2023-yil 11-sentabrdagi "“O'zbekiston - 2030” strategiyasi to'g'risida"gi PF-158-son Farmoni.

14. O'zbekiston Respublikasi Prezidentining 2024-yil 21-iyundagi "Aholi va davlat xizmatchilarining korrupsiyaga qarshi kurashish sohasidagi bilimlarini uzluksiz oshirish tizimini joriy qilish chora-tadbirlari to'g'risida" PQ-228-son Qarori.

III. Maxsus adabiyotlar

1. Schmidt W., Völschow M. // Numerical Python in Astronomy and Astrophysics. A Practical Guide to Astrophysical Problem Solving, Springer, 2021. — 257 p. — ISBN 978-3-030-70346-2
2. T.Axmadjonov, M. Akramov // Fizik jarayonlarni kompyuterda modellashtirish // T.: “Universitet”, 2024
3. M.Akramov, T.Axmadjanov // Fizik jarayonlarni kompyuterda modellashtirish, o‘ quv qo‘llanma, T., 2024
4. Нагаева И.А., Кузнецов И.А. // Основы математического моделирования и численные методы, учебное пособие, Лань.:, М. 2024, 204 с.
5. Слабнов В. Д. // Численные методы, учебное пособие, Лань.:, М. 2024, 392 с.
6. Грабовский, Р.И. Курс физики: Учебник / Р.И. Грабовский. – СПб.: Лань, 2012. – 608 с.
7. Фаритов, Т.А. Курс общей физики: Учебник / Т.А. Фаритов. – СПб.: Лан П, 2016. – 656 с.

IV. Elektron ta’lim resurslari

1. www.edu.uz.
2. www.aci.uz.
3. www.ictcouncil.gov.uz.
4. www.lib.bimm.uz
5. [www. Ziyonet. Uz](http://www.Ziyonet.Uz)
6. www.sciencedirect.com
7. <http://www.msu.ru/>

NAZARIY MASHG'ULOTLAR MAZMUNI

Mavzu 1. Model va modellashtirish tushunchasi. Modellar turlari va klassifikatsiyasi. Kompyuterda modellashtirish. Xisoblash fizikasi. Modellashtirish fizikasi. Kompyuterli eksperiment bosqichlari.

Reja:

1. Modul maqsadi va vazifalari
2. Model va modellashtirish tushunchasi.
3. Modellar turlari va klassifikatsiyasi.
4. Kompyuterda modellashtirish. Xisoblash fizikasi. Modellashtirish fizikasi.
5. Kompyuterli eksperiment bosqichlari.
6. Kompyuterda, modellashtirish uchun operatsion tizim.
7. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for data science.

Tayanch iboralar: *model, modellashtirish, kompyuter modellashtirish, matematik modellashtirish, jarayon, tizim, obyekt, ilmiy tadqiqot, fizik qonunlar, ekologik jarayonlar, iqtisodiy jarayonlar, matematika, fizika, axborot texnologiyalari, tuzilma modeli, funktsional model, statik model, dinamik model, deterministik model, stoxastik model, o'xshashlik nazariyasi, analitik geometriya, differensial tenglamalar, hisoblash algoritmi, dasturlash tili, sonli usullar, matematik statistika, kompyuter dasturi, eksperiment, xatolarni to'g'rilash.*

Hozirgi vaqtda ilmiy va amaliy sohalarda kompyuter modellashtirish turli tizimlar, jarayonlar va atrofimizdagi dunyo hodisalarini o'rganishning asosiy vositalaridan biri hisoblanadi. Biroq, modellashtirish ilmiy tadqiqotlarda uzoq vaqtdan beri qo'llanilib kelinmoqda va asta-sekin fanlarning turli yo'nalishlarini qamrab olgan: qurilish va arxitektura, texnik loyihalash, fizika, astronomiya, kimyo, biologiya va nihoyat, ijtimoiy fanlar. Ammo XX asrning ikkinchi yarmida kompyuterlarning paydo bo'lishi modellashtirishni ilmiy bilishning universal metodiga aylantirdi.

Ma'ruzada asosan kompyuter matematik modellashtirish muammolari ko'rib chiqiladi hamda inson faoliyatining turli sohalariga oid amaliy masalalar bo'yicha matematik va kompyuter modellari yaratiladi.

Fizik jarayonlarni modellashtirishda fizika qonunlarini, ekologik jarayonlarni modellashtirishda biologiya qonunlarini, iqtisodiy jarayonlarni modellashtirishda esa iqtisod qonunlarini bilish zarur. Shuningdek, kompyuter matematik modellashtirish zamonaviy matematik bo'limlardan, jumladan, differensial tenglamalar nazariyasi, analitik geometriya, matematik statistika va sonli metodlarni o'z ichiga oladi. Shu tariqa, kompyuter matematik modellashtirishni o'rganish axborot texnologiyalari bilan tabiiy fanlar, xususan, matematika va fizika o'rtasidagi bog'liqlikni tushunishga yordam beradi.

1. Model tushunchasi

"Model" so'zi lotincha "modulus" so'zidan kelib chiqqan bo'lib, "namuna" ma'nosini anglatadi. U inson faoliyatining turli sohalarida ishlatiladi va turli ma'nolarga ega. Dastlab bu atama qurilish san'ati bilan bog'liq bo'lib, Yevropa tillarida prototip yoki boshqa ob'ekt bilan o'xshash jihatlarga ega bo'lgan ob'ektni ifodalash uchun ishlatilgan.

Kompyuterlarning paydo bo'lishi bilan "model" tushunchasi keng qo'llanila boshlandi, ammo modellar qadim zamonlardan beri insoniyat tomonidan ishlatilgan. Aynan modellar yordamida insonlar muloqot tillarini, yozuvni va grafikani rivojlantirgan. Masalan, qadimgi odamlarning toshga chizilgan rasmlari, haykalchalar, xaritalar, rasmlar va kitoblar atrof-muhit haqidagi bilimlarni keyingi avlodlarga yetkazishning model shakllari hisoblanadi.

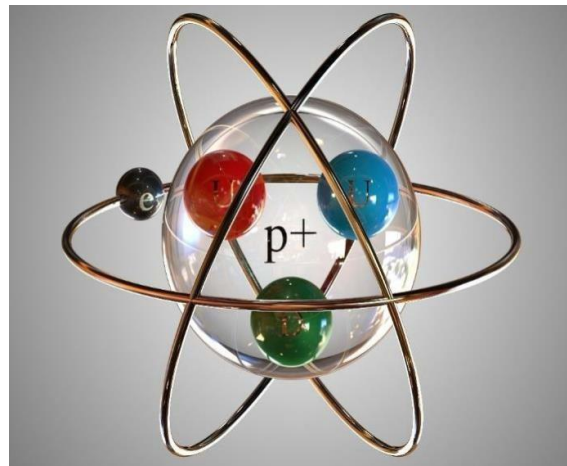


Rasm 1 - Qadimgi modellar - g'or rasmlari

Shuningdek, modellar nafaqat moddiy (tabiiy) shaklda, balki inson tafakkurida ham aks etishi mumkin. Bu inson ongida mavjud bo'lgan dunyo haqidagi ilk tasavvurlarni anglatadi. Kompyuterlarning paydo bo'lishi bu jarayonga yangi bosqich olib keldi: odamlar real ob'ekt yoki jarayonning kompyuter modelini yaratish orqali o'z bilimlarini kompyuterga o'tkaza boshladi. Shunday qilib, dunyoning aks ettirilishi uch bosqichdan iborat:

1. **Birinch**i bosqich – moddiy dunyo (jonli va jonsiz tabiat).
2. **Ikkinchi** bosqich – inson tafakkuri.
3. **Uchinchi** bosqich – jonsiz modda (kompyuter muhiti).

Modellarga insonlar hamma joyda duch keladi. Masalan, bolalikda o'yinchoqlar – plastmassa hayvonlar, qo'g'irchoqlar, mexanik mashinalar, robotlar – haqiqiy ob'ektlarning modellaridir. Ular yordamida bolalar atrof-muhitni o'rganadi.



Modellashtirish nima uchun kerak?

Modellarni yaratishning asosiy sabablari quyidagilardir:

1. **Haqiqiy ob’ekt mavjud emas** – masalan, yo‘qolib ketgan hayvonlar (dinozavrlar), qadimgi tsivilizatsiyalar yoki fantastik mavjudotlar (begonalar, kentavrlar).
2. **Ob’ektni tadqiq qilish qimmat yoki xavfli** – masalan, yadro sinovlari, qora tuynuklar, inson salomatligiga og‘ir sharoitlarning ta’siri, Yer yadrosining harorati.
3. **Ob’ektni tadqiq qilish qiyin** – masalan, Galaktika juda katta o‘lchamga ega, atom yadrosi esa juda kichik.
4. **Faqat ayrim xususiyatlarni o‘rganish zarur** – masalan, ko‘prikning yuk ko‘tarish chegarasi yoki avtomobilning maksimal tezligi.

Modellarning tasnifi

Modellar turli mezonlar bo‘yicha tasniflanadi:

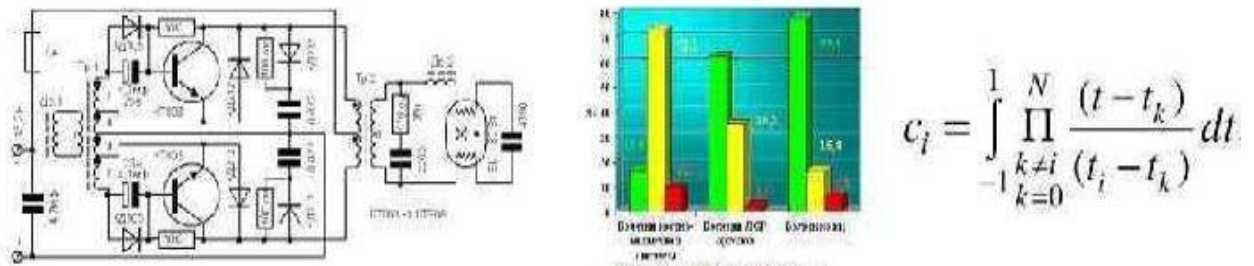
1. **Moddaning mavjudlik shakliga ko‘ra:**
 - **Moddiy modellar** – real dunyodagi ob’ektlarning kichik yoki kattalashtirilgan nusxalari (maketlar, fizik modellar).
 - **Axborot (abstrakt) modellar** – inson tafakkurida yoki belgilar yordamida ifodalanadigan modellar (grafikalar, sxemalar, formulalar).
2. **Vaqt faktori bo‘yicha:**
 - **Statik modellar** – ob’ektning muayyan vaqt ichidagi holatini aks ettiradi (xaritalar, skeletlar).
 - **Dinamik modellar** – ob’ektning vaqt bo‘yicha o‘zgarishini tasvirlaydi (harakat tenglamalari, meteorologik prognozlar).
3. **Jarayonning tabiatiga ko‘ra:**

- **Deterministik modellar** – tasodifiylik ta’sir qilmaydigan jarayonlarni tasvirlaydi (fizik formulalar).
- **Stoxastik modellar** – tasodifiy hodisalar hisobga olinadigan modellar (iqtisodiy tahlillar, Brown harakati).

4. Ob’ektning xususiyatlariga ko‘ra:

- **Tuzilish modeli** – ob’ektning ichki tuzilishini tasvirlaydi.
- **Tashqi ko‘rinish modeli** – ob’ektning tashqi shaklini aks ettiradi.
- **Funksional model** – ob’ekt qanday ishlashini tushuntiradi.
- **Xulq-atvor modeli** – ob’ektning qanday harakat qilishini tasvirlaydi.

Shuningdek, **belgili modellar** (grafiklar, formulalar, matnli tasvirlar) va **matematik modellar** (differensial tenglamalar, hisob-kitob usullari) ham modellashtirish jarayonida keng qo‘llaniladi.



Rasm 4 - Belgili modellar (sxema, diagramma, formula)

Modellashtirish atrof-muhitni tushunish va tadqiq qilishda muhim vosita hisoblanadi. U insonlar vaqt va resurslarni tejashga, xavfsizroq va samaraliroq tajribalar o‘tkazishga imkon beradi. Zamonaviy ilm-fanda modellashtirish deyarli barcha sohalarda muhim ahamiyat kasb etadi.

Kompyuterda modellashtirish. Xisoblash fizikasi. Modellashtirish bosqichlari

Kompyuterda modellashtirish bu yangi va yetarlicha murakkab kurs. Uni yaxshi o‘zlashtirish uchun bir necha bilimlar talab qilinadi: birinchidan, tanlangan

fan sohasi bo'yicha bilimlar – agar biz fizik jarayonlarni modellashtirayotgan bo'lsak, biz kerakli darajadagi fizika qonunlari bilimlarini egallagan bo'lishimiz, ekologik jarayonlarni modellashtirishda – biologiya qonunlarini, iqtisodiy jarayonlarni modellashtirishda – iqtisod qonunlarini bilishimiz, bundan tashqari kompyuterda modellashtirish amalda barcha zamonaviy matematik apparatlarni qo'llab amalga oshirilishini inobatga olsak, matematik bilimlar zarur bo'ladi.

Kompyuterda matematik masalalarni yechish uchun noxiziqli tenglamalarni sonli yechish, chiziqli tenglamalar sistemasini, differensial tenglamalarni yechish usullarini va funksiyalarni tekshirish usullarini bilishi talab etiladi. Shuningdek, albatta, zamonaviy axborot texnologiyalaridan erkin foydala olinishi va dasturlash tillarini bilishi hamda amaliy dasturlardan foydalana olish ko'nikmasiga ega bo'lishi kerak.

Nazariy va eksperimental fizika bilan bir qatorda hozirgi kunda hisobli (kompyuterli) fizika* sohasi ham mavjuddir. Ushbu soha nazariy fizikaning hisobli tahlilga asoslangan bo'limi rivojlanishi va mukammallashuvi, eksperimental fizika sohasida zamonaviy kompyuterlarni tajribani boshqarish va o'lchovlarni o'tkazish, o'lchash natijalarini hisoblash uchun samarali qo'llash natijasida shakllandi. Kompyuterli fizikadagi ilmiy tadqiqotlar yangi texnologiya va uslubiyatga asoslangan holda olib boriladi. Shuni aytish lozimki, hozirgi davrda kompyuter faqatgina hisoblash amallarini tezlatuvchi, talabalar bilimini tekshiruvchi vositagina bo'lmay, o'qitishni yakkama-yakka amalga oshiruvchi va eng asosiysi - fizik jarayonlarning modelini yaratuvchi vositaga ham aylandi. Bunda kompyuter yordamida jiddiy muammolarni yechish bosqichlaridan iborat texnologik siklni o'z ichiga olgan yetarlicha murakkab bo'lgan ilmiy-ishlab chiqarish jarayonini talab etadi:

1. Masalaning qo'yilishi.
2. Formallashtirish (matematik modelni yaratish).
3. Hisoblash algoritmini ishlab chiqish.
4. Kompyuter dasturini ishlab chiqish.

5. Hisoblash amallari.
6. Dasturni sozlash.
7. Natijalarni olish va tahlil qilish
8. Xatolarni to'g'rilash.

Kompyuterli texnologiyaning rivojlanishi natijasida fizik tizimlarga yangicha qarash shakllandi. Dolzarb muammolarni kompyuter vositasida hal etishda ilmiy qonunlarni faqat differensial tenglamalar bilangina emas, balki kompyuter uchun yozilgan qoidalar tarzida ham ifodalash qulay ekanligi ayon bo'ldi. Fizik jarayonlarni o'rganishga bunday yondashish fiziklarning kompyuterga bo'lgan munosabatini o'zgartirdi. Endi kompyuterlar tabiiy jarayonlarni modellashtiruvchi ma'lum fizik tizim sifatida ko'rilmogda.

Kompyuterli modellashtirish jarayoni laboratoriya eksperimentiga o'xshash, shuning uchun ham u ba'zan **kompyuterli eksperiment*** deb ham ataladi. Quyidagi jadvalda ularning o'xshash xususiyatlari keltirilgan:

Laboratoriya eksperimenti	Kompyuterli eksperiment
Fizik jarayon	Model
Fizik asbob	Kompyuter dasturi
Kalibrovka	Dasturni rostdash
O'lchash	Hisoblash
Natijalar tahlili	Natijalar tahlili

Kompyuter uchun tuzilgan dastur fizik jarayonni modellashtirgan holda kompyuterli eksperimentni o'zida aks ettiradi. Bunday eksperiment, odatda, laboratoriya eksperimenti deyiladi, hamda nazariy hisob-kitoblar orasida «ko'prik» bo'lib xizmat qiladi. Xususan, ideallashtirilgan modelning kompyuterli modelidan foydalangan holda aniq natijalar olishimiz mumkin. Vaholanki, bunday mavhum modelni laboratoriya sharoitida umuman yaratib bo'lmaydi. Shu bilan birga, real

model asosida olib borilgan kompyuterli eksperiment natijalarini bevosita laboratoriya eksperimenti natijalari bilan taqqoslash mumkin.

Shuni ta'kidlab o'tish mumkinki, kompyuterli modellashtirish fikrlash jarayonining o'rnini bosmaydi, balki laboratoriya eksperimenti kabi murakkab hodisalarning mohiyatini ochib berishda qurol sifatida ishlatiladi.

Endi kompyuterli eksperiment jarayoniga xos bo'lgan bosqichlarning asosiy xususiyatlarini ko'rib chiqaylik.

Birinchi bosqich – masalaning qo'yilishi. Bu bosqichda masala bayon etiladi, uni yechish maqsadi qo'yiladi, kiruvchi va chiquvchi axborotlar tahlil qilinadi, masalaning mohiyati og'zaki ifodalanadi va uni yechishga umumiy yondoshish bo'yicha fikr beriladi. Aniq predmet sohasidagi malakali mutaxassis asosan masalaning qo'yishni amalga oshiradi.

Ikkinchi bosqich – formallashtirish (rasmiylashtirish). Uning maqsadi - masalaning, kompyuterda adekvatlikni yo'qolmasdan ishlatish mumkin bo'lgan, matematik modelini yaratishdir. Agar masala murakkab bo'lmasa va maxsus matematik bilimni talab qilmasa bu bosqichni masala qo'yuvchining o'zi bajarishi mumkin, aks holda bu ishga matematik yoki dasturchini jalb qilish maqsadga muvofiq bo'ladi.

Ma'lum fizik jarayon yoki hodisa sonli kattaliklar yordamida ifodalangan taqdirdagina uning tavsifi ishonchli bo'lishi Galiley zamonidan buyon ma'lum. Bunday kattaliklarning bir qismi tajribada o'lchanadi, qolgan qismini aniqlash uchun esa matematik masalalar shakllantiriladi. Fizika nazariyalarini matematik tarzda ifodalash zaruriyati e'tirof etilgandan so'ng, real borliqni tavsiflash eksperiment va nazariya orasidagi o'zaro ta'sirlar ketma-ketligiga aylandi. Nazariyaning maqsadi - eksperimentning qoniqarli matematik ifodasini izlashdan iborat. Bunda nazariya qator fundamental tamoyillarga (termodinamika tamoyillari, saqlanish qonunlari, invariantlik va h.k.) asoslanib, matematik apparat yordamida bu tamoyillardan bashorat etish uchun zarur bo'lgan axborotni olishga intiladi.

Klassik fizika bashorat etish imkoniyatiga ega bo'lgan nazariyalarga asoslangan edi. Davr o'tishi bilan nazariya kuzatilayotgan hodisalarni bilish vositasi

sifatida tan olindi. Hozirgi vaqtda har qanday nazariyaning ahamiyati cheklangan aksiomatik fikrlar yordamida o‘zaro bog‘lanmagan ko‘p sonli faktlarni bayon etish imkoniyati bilan baholanadi. Shuni ta’kidlash joizki, zamonaviy kompyuterlar ixtiro qilinguncha real borliqni nazariy tavsiflash darajasi, ya’ni matematik modellarning murakkablik darajasi ularga mos keluvchi matematik masalalarni yechish imkoniyatlaridan sezilarli ilgarilab ketgan edi. Masalan, Butun Olam tortishish qonunining kashf etilishi bilanoq N ta jism haqidagi masalani ifoda etish mumkin bo‘ldi. Bunday masala N ta o‘zaro ta’sirlashayotgan moddiy nuqtaning vaqt bo‘yicha o‘zgarishini o‘rganishga bag‘ishlangan. Garchi fizik jarayonning matematik modeli yaratilib, masala to‘g‘ri ifoda etilgan bo‘lsa-da, cheksiz katta hajmdagi hisoblash amallari tufayli ushbu jarayonni to‘g‘ri tahlil etish imkoniyati yo‘q edi.

Aksariyat fizik hodisalar ma’lum kattaliklar hamda kattalikning o‘zgarish koeffitsiyentlari orasidagi munosabatlar vositasida ta’riflanadi. Masalan, dinamikaning asosiy qonuni

$$\vec{F} = m\vec{a} \quad (1.1)$$

harakatlanayotgan jism tezligining vaqt bo‘yicha o‘zgarishini unga ta’sir etayotgan kuch bilan bog‘lanishini ifodalaydi. Agar bu U potensial tufayli yuzaga kelgan bo‘lsa, u holda

$$\vec{F} = \nabla U \quad (1.2)$$

Bu ifodada kuch U funksiyaning fazo bo‘yicha o‘zgarishini aks ettiruvchi operator orqali bog‘langan. Matematik amallar ushbu munosabatlarni differensial tenglamalarga o‘zgartiradi. Ko‘p hollarda zarur matematik masalalarning analitik yechimini hosil qilish mumkin bo‘lmaydi, chunki izlanayotgan yechim elementar yoki boshqa ma’lum funksiyalar vositasida ifodalanmaydi. Vaholanki, transsendent yoki trigonometrik funksiyalar vositasida hosil qilinuvchi analitik yechim mavjud bo‘lsa, hisoblash algoritmlarini tuzishni birmuncha yengillashtirgan bo‘lar edi. Afsuski, aksariyat fizik hodisalarning matematik taqlidi differensial tenglamalar va ba’zan xususiy hosilali tenglamalarning yechimi bilan bog‘liq bo‘ladi. Haqiqiy

o‘zgaruvchili va xususiy hosilali tenglamalar nazariyasiga ko‘ra ular asosan uch toifaga bo‘linadi:

1. Giperbolik tenglamalar

$$\frac{\partial^2 V}{\partial t^2} = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = \nabla^2 V \quad (1.3)$$

Ushbu turdagi tenglamalar to‘lqinlarning tarqalishidagi fizik jarayonlarni tavsiflaydi.

2. Parabolik tenglamalar

$$\frac{\partial V}{\partial t} = \nabla^2 V \quad (1.4)$$

Issiqlik, gazlar, suyuqliklar hamda elektromagnit maydondagi diffuziya hodisalari bunday tenglamalar yordamida tavsiflanadi.

Elliptik tenglamalar

$$\nabla^2 V = 0 \quad (1.5)$$

parabolik tenglamalarning $t \rightarrow \infty$ holdagi asimptotik statsionar holatini ifodalaydi. Bunday masalalar siqilmaydigan suyuqlik (yoki gaz) yoki elektr tokining statsionar holatini, elektr zaryadlarining yoki issiqlik manbasi bilan bog‘langan jismning muvozanat holatini tavsiflaydi.

Ixtiyoriy ikkinchi tartibli differensial tenglamani yuqorida ko‘rsatilgan toifadagi tenglamalarning biriga keltirish mumkin. Fazo va vaqt o‘lchamligi shunday tanlanadiki, tenglamaga kiruvchi koeffitsiyentlar birga teng bo‘lishi lozim.

Differensial tenglamalar yordamida taqlid etiluvchi fizik hodisalarning xilma-xillik xususiyati umumlashtirilgan matematik modellarni yaratishni mushkullashtiradi. Shuning uchun bunday tenglamalarni kompyuter yordamida yechish jarayonida fizik-tadqiqotchi ularning fizik ma‘nosi hamda matematik mazmunini bir vaqtning o‘zida talqin etishi lozim.

Muammoning fizik mohiyatini hamda matematik modelini o‘zaro uyg‘unlashtirilgan holda ifoda qilish uning to‘g‘ri yechimini aniqlash garovidir.

Uchinchi bosqich – hisoblash algoritmini ishlab chiqish. Kompyuterli eksperimentning ikkinchi bosqichi ifoda qilingan matematik masalaning yechish uslubini ishlab chiqishdan iborat. Bunda tahliliy va hisoblash usullaridan oqilona foydalangan holda bir nechta algebrik tenglamalar va ulardan qaysi ketma-ketlikda foydalanish qoidalari ishlab chiqiladi. Hosil qilingan algoritm tadqiq etilayotgan fizik jarayonni aks ettiruvchi differensial tenglamani yechishga, ya'ni fizik kattaliklarni aks ettiruvchi parametrlarning ma'lum qiymatlarida uni bevosita hisoblashga mo'ljallangan bo'ladi.

Eksperiment kabi sonli model* ham ma'lum fizik asosga tayangan holda ishlab chiqiladi. Odatda, fizik eksperiment biz anglamoqchi bo'lgan borliqning ma'lum modeli sifatida namoyon bo'ladi. Agar bu voqelik nihoyatda murakkab bo'lib, eksperiment o'tkazishga imkoniyat bo'lmasa, tabiiyki, biz nisbatan sodda eksperimental model yaratishga intilamiz. Demak, aksariyat fizik eksperimentlar to'laligicha muammoni emas, balki uning xususiy modellaridan birini o'rganishga xizmat qiladi. Sonli model shunday eksperimentlarni eslatadi.

Sonli modelni yaratishda dastlab ma'lum bir fizik vaziyatni tavsiflovchi qonunlarni kompyuter vositasida taqdim etish lozim. Hodisa yetarlicha murakkab bo'lsa, olingan natijalar kutilayotgan natijalardan farqli bo'ladi. Bunday holda tadqiqotchi o'rganayotgan hodisani har tomonlama tahlil etib, sinchiklab o'rganishi lozim.

Eksperiment jaryonida bunday tahlil o'lchashlarga, sonli tahlilda esa oraliq natijalar hamda yordamchi kattaliklarga asoslanadi. Eksperiment to'g'risidagi umumiy tushunchalar shakllanishi bilanoq uni samarali amalga oshirish uchun zarur bo'lgan tahlil haqida ham mulohaza yuritish lozim. Shuningdek, ma'lum dasturni ishlab chiqishda muhim yordamchi kattaliklarga murojaat etish imkoniyatini ham hisobga olish zarur. O'lchash amallarisiz olib borilgan tajriba kabi natijasiz dastur ham foydasizdir. Shunday qilib, hisoblash algoritmi goh nazariyaning quroli, goh eksperimentning yangi turi bo'lib xizmat qiladi. Kompyuter vositasida hisoblash algoritmini yechish ham, matematik model ham, aslida nazariyaning bir xil ahamiyatga ega bo'lgan tarkibiy qismlari sifatida talqin etilishi kerak.

Sonli modelning afzalliklari va noqulayliklari xususida quyidagilarni aytish mumkin. Hisoblash vositalarining faqat modellar tarzida namoyon bo'lishi ularning eksperimentga nisbatan noqulayligini ko'rsatadi. Shuning uchun ham natijalarning nazariyani taqlid etuvchi qismini hamda sonli modelning xususiyatlari tomonidan taqlid etilgan qismini bir-biridan ajrata bilish zarur.

Sonli model quyidagi ikki ajoyib xossaga ega: raqam usulida olingan natijalarni takroran olish mumkin (hattoki ixtiyoriy tasodifiy jarayonlarni modellashtirganda ham); sonli modellarning bashorat etish imkoniyatlari eksperimentga nisbatan yuqoriroq. Darhaqiqat, ixtiyoriy momentda sonli modelning barcha tafsilotlari ma'lum bo'ladi; ularni bilish uchun modelni o'zgartirish talab etilmaydi. Vaholanki, fizik eksperimentda bunday imkoniyat mavjud emas.

Endi qanday tarzda sonli modelni tadbiq etish lozimligi haqida mulohaza yuritamiz. Dastlab uni nazariya va eksperiment bilan taqqoslash kerak. Sonli model to'g'ri nazariy modelni talab etadi. Agar fizik jarayonning matematik modeli noto'g'ri ishlab chiqilgan bo'lsa, masalaning yechimini kompyuterda to'g'rilab bo'lmaydi. Nazariy modelning yechimi mavjud bo'lsa, analitik usulda zaruriy natijalar olinadi. Biroq, fizik jarayonlarni aks ettiruvchi barcha tenglamalar bunday yechimga ega emas. Bunday hollarda amaliy matematikaning hisoblash uslublaridan oqilona foydalanish zarur. Bunda eksperiment, dastur tuzish hamda hisoblash amallari bilan bog'liq bo'lgan qiyinchiliklarni hisobga olish zarur.

Murakkab eksperimentlardagi kam o'rganilgan hodisalarni bashorat etishda sonli model ayniqsa foydalidir. Bunday maqsadlarda kompyuter amaliy fizika sohalarida tobora kengroq ko'lamda qo'llaniladi. Shuni ta'kidlash joizki, kompyuterdan foydalanish sohasidagi har qanday rivojlanish, bilish darajamizning sezilarli siljishiga olib keladi; va aksincha, ma'lum jarayonlarni modellashtirish va ularda o'ta aniq o'lchashlar olib borish imkoniyatlari shu jarayonlarning matematik tavsifini qayta ko'rib chiqishni taqozo etadi.

To'rtinchi bosqich – kompyuter dasturini ishlab chiqish. Bu bosqichda dastur tuziladi. Dastur - kompyuter tushunadigan tilda yozilgan algoritmni ifodalash shaklidir. Algoritm matn yoki grafik ko'rinishdagi inson tushunadigan

tilda, dastur esa maxsus algoritmik tilda yoziladi. Dasturni tuzish jarayonida algoritmgaga yanada aniqlik kiritish mumkin.

Avvalgi bo'limlarda bayon etilgan mulohazalarga asoslangan holda, ma'lum bir fizik jarayon matematik tarzda ifoda etilgan hamda uning sonli modeli yaratilgan bo'lsin. Izlanayotgan yechimning xossalari ma'lum bo'lsa, taxminiy algoritmik yechimlar dasturni tekshirish imkonini beradi. Demak, navbatdagi bosqichda ana shu algoritmnini yuqori saviyadagi dasturli tilda yozish lozim.

Shuni aytish joizki, aynan bir jarayonni, garchi uning matematik ifodalari va hisoblash uslublari aniqlangan bo'lsa-da, amalda turli xil dasturlar vositasida modellashtirish mumkin. Dasturning barcha variantlaridan eng samaralisini tanlash tadqiqotchining kompyuter bilan muloqot qilishida ancha yengillik yaratib beradi. Shuning uchun dasturni oqilona ishlab chiqishda quyidagi mezonlarni hisobga olish zarur.

a) dasturning modulliligi. Bir necha modullardan iborat bo'lgan dastur qator ijobiy xossalarga ega. Xususan, operatorlar miqdori qisman o'zgarganda xatolarni aniqlash ancha osonlashadi; dasturning boshqa qismlarini o'zgartirmagan holda faqat bir qismini takomillashtirish yoki o'zgartirish mumkin (masalan, bir dasturni boshqasiga almashtirsa bo'ladi).

Sonli model bilan ishlash jarayonida dasturda hisobga olinmagan yangi fizik hodisalarni tavsiflash uchun ma'lum bir tavsillarni o'zgartirish zaruriyati paydo bo'ladi. Agar dastur modulli usulda yozilgan bo'lsa, bunday hollarda dasturning bir qismi o'zgartiriladi. Yuqori saviyali tilda yozilgan modulli dasturlardan boshqa soha mutaxassislari ham osonlikcha foydalanishlari mumkin.

b) o'zgaruvchilarni tanlash. Garchi o'zgaruvchilar modellashtirilayotgan masalaga bog'liq bo'lsa-da, tadqiqotchi ularning nomini tanlashda va ularning tuzilishini tashkil etishda ma'lum erkinlikka ega. O'zgaruvchining nomini tanlash oson bo'lmagan masaladir. Fizikada ko'p uchraydigan kattaliklarni taqdim etishda ma'lum ifodalardan foydalanish zarur: vaqt – t , entropiya - s , oqim- I va h.k. O'zgaruvchilarning nomini ham shunday tanlash kerakki, ular ifoda etilayotgan fizik

kattalikni eslatishi lozim. Shunda buyruqlarni izohlash va xatolarni izlash kabi amallar osonlashadi.

v) matematik ifodalarni yozish. Yetarlicha murakkab bo'lgan matematik ifodalarni bo'laklab, oddiy hisoblashlar ketma-ketligi tarzida ifodalash lozim. Biroq, bunday shaklda ifodalangan amallar majmuasini talqin etish, yozish va ularning xatolarini aniqlashda birmuncha qiyinchiliklar tug'iladi.

g) ma'lumotlarni kiritish va chiqarish. Bunday buyruqlardan oqilona foydalanish dasturdan tadqiqot quroli sifatida unumli foydalanish imkoniyatini yaratib beradi. Shuning uchun dasturning ishlashini kuzatish imkonini beruvchi hamda yechimning aniq qiymatlarini ifoda etuvchi ma'lumotlar, fizik kattaliklar va axborotlarni oydin holda aks ettirish lozim. Yagona parametrning qiymatini aniqlash lozim bo'lgan hollarda ham modelni to'laligicha kuzatish, va demak, birmuncha natijalarni ham nazarda tutmoq maqsadga muvofiqdir. Bunday natijalar eksperimentning diagnostikasi vazifasini bajarib, dasturdan foydalanishda hamda modelning asosiy gipotezalari to'g'ri ekanligi haqida xulosa chiqarishda muhim ahamiyatga ega.

Beshinchi bosqich – hisoblash amallari.

Oltinchi bosqich – dasturni sozlash.

Oxirgi ikki bosqich o'zaro bog'langan. Ularda dasturning to'g'ri ishlashi tekshiriladi. Shu maqsadda modellashtiriladigan masaladagi hamma holatlarni imkoni boricha e'tiborga oluvchi test misoli tuziladi. Avvaldan ma'lum test misoli natijasiga olingan natijaning mos kelishiga qarab dasturning to'g'ri ishlashi baholanadi. Dastur tuzilib, tekshirilgandan keyin uni qo'llash mumkin.

Yettinchi bosqich – natijalarni olish va tahlil qilish. Bu yerda masalani qo'yuvchi tomonidan yechim natijasi tahlil qilinadi va boshqarish qarorlari yoki takliflari qabul qilinadi.

Sakkizinchi bosqich – xatolarni to'g'rilash (korrektirovka). Agar dasturni qo'llashda qoniqarsiz natija olinsa, model va algoritmgaga tuzatishlar kiritish talab qilinadi. Bu bosqichning bajarilishi oldingi bosqichlarning ixtiyoriysini tuzatish, mukammallashtirish zarurati bilan bog'liq.

Kompyuterda modellashtirish uchun operatsion tizim.

Dasturiy ta'minot

Kompyuterli eksperimentning dasturiy ta'minotini kompyuterning ajralmas qismi bo'lgan algoritmik til, translyator, operatsion sistema, standart dasturlar kutubxonasi hamda turli xildagi amaliy dasturlar majmuasi tashkil etadi. Amaliy dasturlar majmuasi, odatda, universal bo'lmay, balki ma'lum toifadagi masalalarni yechish uchun ishlab chiqilgan bo'ladi. Shu bilan birga, fizik jarayonlarning aksariyati bir xil turdagi tenglamalar yordamida ifodalanishi mumkin. Masalan, issiqlik o'tkazuvchanlik tenglamasi issiqlikning uzatilishi bilan birga o'tkazgichda magnit maydonining tarqalishi yoki zarralar diffuziyasi kabi hodisalarni ham miqdoriy tavsiflaydi. Shuning uchun kompyuterli eksperimentning dasturiy ta'minotini yaratishda mavjud amaliy dasturlar majmuasidan unumli foydalanish lozim.

Operatsion tizimlar*. Operatsion tizim (OT) nima? Operatsion tizim kompyuter ishga tushirilishi bilan yuklanuvchi shunday bir dasturki, bu dastur foydalanuvchiga kompyuter bilan muloqot qilish vositasi bo'lib xizmat qiladi, uning barcha qurilmalari ishini boshqarish imkonini beradi. Operatsion tizim yordamida tezkor xotiradan foydalanish, disklardagi ma'lumotlarni o'qish yoki ma'lumotlarni disklarga yig'ish, fayllarni ko'chirish yoki bosmaga chiqarish amaliy dasturlarni tez yuklab ishga tushirish va boshqaruvni ularga uzatish, operativ xotirani dastur ishi so'ngida bo'shatib yana o'ziga olish kabi ishlar bajariladi.

OT – kompyuterda ishlovchi eng asosiy dasturiy ta'minot, inson va kompyuter orasidagi muloqot “ko'prigi”. Operatsion tizimsiz kompyuter ishlamaydi. Operatsion tizim kompyuterga o'rnatilgan bo'ladi va uni zarur bo'lganda o'zgartirish mumkin. Microsoft Windows, Apple Mac OS X i Linux eng keng tarqalgan operatsion tizimlar hisoblanadi. Kompyuterda modellashtirish uchun qaysi operatsion tizim yaxshi? Bu savolga javob qo'yilgan va yechilayotgan masalaga bog'liq:

Alohida kompyuterlar (PC)	Windows, Linux, Mac OS
Klasterlar	Linux

Superkompyuterlar	Unix, Linux
Hisobga olinishi kerak bo'lgan parametrlar:	mavjud qurilma va jixozlar; dasturiy ta'minot; stabillik; natijalar tahlili va taqdimot

Fizikada keng qo'llanadigan dasturlash tillari: Fortran, C/C++, Java. Qaysi dasturlash tilini ishlatish kerak?

Tanlov qo'yilgan va yechilayotgan masalaga bog'liq:

- sonli modellashtirish;
- tizimli dasturlash;
- Web dasturlash;

mavjud dasturiy kutubxona va kodlar
tajriba, malaka.

[Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for Data Science](#)

Zamonaviy ilm-fan va muhandislik sohalarida ma'lumotlarni qayta ishlash, hisoblash va modellashtirish muhim ahamiyat kasb etadi. Buning uchun eng keng tarqalgan dasturiy vositalar qatoriga **Spyder**, **MatlabWork**, va **Python for Data Science** kiradi. Ushbu ma'ruzada ushbu dasturlar haqida tushuncha berilib, ularni o'rnatish yo'llari ko'rsatiladi.

Spyder (Scientific Python Development Environment) - bu **Python** dasturlash tilida ilmiy hisob-kitoblar va ma'lumotlarni tahlil qilish uchun mo'ljallangan ochiq kodli muhitdir. U Python'ning **NumPy**, **SciPy**, **Pandas** va **Matplotlib** kabi mashhur kutubxonalari bilan integratsiyalashgan bo'lib, Data Science va muhandislik hisob-kitoblari uchun keng qo'llaniladi.

Spyder'ning xususiyatlari

- **Interaktiv dasturlash muhiti** - Matn muharriri, konsol va to'plangan ma'lumotlarni ko'rish imkoniyati.
- **Ilmiy hisob-kitoblar uchun moslashuvchan** - MATLAB'ga o'xshash qulay interfeys.
- **Kod yozish va debugging** - Intellisense, avtomatik kod to'ldirish va hatoliklarni tekshirish.
- **Vizualizatsiya vositalari** - Ma'lumotlarni grafik va jadval shaklida ko'rsatish.

Spyder Python muhitining **Anaconda** to'plami tarkibida mavjud. Uni o'rnatish uchun quyidagi amallar bajariladi:

1. Anaconda'ni yuklab olish:

- <https://www.anaconda.com/> sahifasidan o'zingizga mos versiyani yuklab oling.

2. Anaconda'ni o'rnatish:

- O'rnatish jarayonida **Spyder** avtomatik ravishda o'rnatiladi.

3. Spyder'ni ishga tushirish:

- Terminal yoki **Anaconda Navigator** orqali **Spyder** dasturini oching.

MatlabWork

MatlabWork - bu **MATLAB dasturiy muhiti** va Python imkoniyatlarini birlashtiruvchi dastur. U ayniqsa muhandislik va ilmiy hisob-kitoblarda, jumladan, signal qayta ishlash, tasvirlarni qayta ishlash va matematik modellashtirishda keng qo'llaniladi.

MatlabWork'ning asosiy xususiyatlari

- **MATLAB va Python integratsiyasi** - MATLAB kodlarini bevosita Python orqali ishga tushirish imkoniyati.

- **Matematik va ilmiy hisob-kitoblar** - Murakkab differensial tenglamalar, algebraik va statistik operatsiyalarni bajarish.
- **Vizualizatsiya va grafik imkoniyatlari** - Diagrammalar va grafik chizish qulayligi.

MatlabWork'ni o'rnatish uchun MATLAB'ning rasmiy veb-saytidan yuklab olinadi: <https://www.mathworks.com/> dan MATLAB'ni yuklab olinadi va o'rnatish faylini ishga tushiriladi. MATLAB va Python'ni bog'lash uchun matlab.engine modulidan foydalaniladi. Python'da quyidagi kod orqali MATLAB'ni ishga tushirish mumkin:

- import matlab.engine
- eng = matlab.engine.start_matlab()
- eng.sqrt(16)

Python for Data Science

Python Data Science uchun eng mashhur dasturlash tillaridan biri bo'lib, uning kuchi turli xil kutubxonalarda namoyon bo'ladi:

- **NumPy** – sonli hisob-kitoblar uchun
- **Pandas** – ma'lumotlarni qayta ishlash
- **Matplotlib va Seaborn** – vizualizatsiya
- **Scikit-learn** – mashinani o'rganish uchun

Python for Data Science'ni o'rnatish (Anaconda orqali) quyidagicha amalga oshiriladi: Anaconda'ni yuklab olish va o'rnatish <https://www.anaconda.com/> dan Anaconda'ni yuklab olinadi. Python muhitini ishga tushirish uchun Terminal yoki Anaconda Navigator orqali jupyter notebook yoki spyderni ishga tushiriladi. Kerakli kutubxonalarni o'rnatiladi:

- pip install numpy pandas matplotlib seaborn scikit-learn
- conda install numpy pandas matplotlib seaborn scikit-learn

Ushbu ma'ruzada **Spyder, MatlabWork va Python for Data Science** dasturlarining umumiy tavsifi va ularni o'rnatish yo'llari ko'rib chiqildi.

Demak Spyder - ilmiy hisob-kitoblar va Python dasturlash uchun kuchli muhit, MatlabWork - MATLAB va Python integratsiyasiga asoslangan muhandislik hisob-kitoblari vositasi va Python for Data Science - ma'lumotlarni tahlil qilish va vizualizatsiya qilish uchun eng qulay vosita hisoblanadi.

Ushbu dasturlar zamonaviy ilmiy tadqiqot va muhandislik muhitida keng qo'llaniladi va ularning imkoniyatlarini o'rganish zamonaviy mutaxassis uchun juda muhimdir.

Nazorat savollari

1. Model nima?
2. Modellashtirish deganda nima tushuniladi?
3. Modellarini qanday sinflarga ajratish mumkin?
4. Modellarga tushuntirish bering.
5. Matematik modellashtirish nima?
6. Kompyuterda modellashtirish nima?
7. Hisoblash fizikasi nima?
8. Imitatsion model nima?
9. Matematik model qurish uchun nimalar kerak?
10. O'rganilayotgan obyekt, jarayon yoki tizimni matematik ifodalash nimalarga bog'liq?
11. Matematik modelni tanlash bosqichida nimalar aniqlanadi?
12. Kompyuterda modellashtirishning asosiy bosqichlarini tushuntiring.
13. Operatsion tizim nima?

14. Spyder dasturiy ta'minoti nima va uning asosiy xususiyatlari qanday?
15. MatlabWork qanday funksiyalarga ega va u qaysi sohalarda qo'llaniladi?
16. Python for Data Science qanday kutubxonalardan foydalanadi?
17. Spyder dasturini Anaconda orqali qanday o'rnatish mumkin?
18. Matlab va Python integratsiyasini qanday amalga oshirish mumkin?
19. Data Science uchun Python dasturlash tilining afzalliklari nimalardan iborat?
20. MatlabWork va Spyder'ning o'zaro farqlari nimada?
21. Python muhitida ma'lumotlarni vizualizatsiya qilish uchun qaysi kutubxonalar ishlatiladi?
22. Anaconda platformasining asosiy ustunliklari nimalardan iborat?
23. Kompyuterda ilmiy hisob-kitoblar va modellashtirish uchun qaysi dasturlarni tavsiya qilasiz?

Foydalanilgan adabiyotlar

1. Schmidt W., Völschow M. // Numerical Python in Astronomy and Astrophysics. A Practical Guide to Astrophysical Problem Solving, Springer, 2021. — 257 p. — ISBN 978-3-030-70346-2
2. T.Axmadjonov, M. Akramov // Fizik jarayonlarni kompyuterda modellashtirish // T.: "Universitet", 2024
3. M.Akramov, T.Axmadjanov // Fizik jarayonlarni kompyuterda modellashtirish, o' quv qo'llanma, T., 2024
4. Нагаева И.А., Кузнецов И.А. // Основы математического моделирования и численные методы, учебное пособие, Лань., М. 2024, 204 с.
5. Слабнов В. Д. // Численные методы, учебное пособие, Лань., М. 2024, 392 с.
6. Грабовский, Р.И. Курс физики: Учебник / Р.И. Грабовский. – СПб.: Лань, 2012. – 608 с.
7. Фаритов, Т.А. Курс общей физики: Учебник / Т.А. Фаритов. – СПб.: Лань, 2016. – 656 с.

Mavzu 2. Kodlarni tuzish: algoritmlar, o'zgaruvchanlar, ma'lumotlar turlari. Python tiliga kirish: projektни saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari. Tarmoqlanuvchi algoritmlar. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash. Mantiqiy operatsiyalar. Sikllar bilan ishlash.

Reja:

1. Kodlarni tuzish: algoritmlar, o'zgaruvchanlar, ma'lumotlar turlari.
2. Python tiliga kirish: projektни saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari.
3. Tarmoqlanuvchi algoritmlar, if va else shart operatorlari bilan ishlash misollari. Chala tarmoqli va to'liq tarmoqli algoritmlar.
4. Munosabatlar, tanlash va solishtirish operatorlari bilan ishlash. Mantiqiy operatsiyalar.
5. Sikllar bilan ishlash: sikllar va siklik operatorlar. Sikllar yordamida summa va ko'paytmalarni topish.

Tayanch iboralar: *Kodlar,algoritmlar, o'zgaruvchanlar, ma'lumotlar turlari? Python tili, projektни saqlash va tayyorini davom etish, sintaksisining asosiy qoidalari.*

Python dasturlash tili



Python - yuqori bosqichli dasturlash tili hisoblanib, tirli xil ilovalarni yaratish uchun mo'ljallangan. Ya'ni Python dasturlash tili yordamida veb-ilovalar, o'yin ilovalari, oddiy (nastol'niy) dasturlar yaratish hamda berilganlar bazasi bilan ishlash mumkin. Ayniqsa Python dasturlash tilining tezlik bilan tarqalishiga uning mashinali o'rgatish va sun'iy intellekt sohalaridagi tadqiqot ishlarida keng qo'llanilishi sabab bo'lgan.

Python dasturlash tiliga 1991 yil Golland dasturchisi Grido Van Rossu asos solgan. Shundan beri ushbu til rivojlanishning ulkan yo'lini bosib o'tdi va 2000 yilda 2.0 versiyasi, 2008 yil esa 0 versiyalari chiqarildi. Versiyalar orasidagi muddatning

uzoqligiga qaramasdan doima versiya ostilari chiqariladi. Shunday qilib, ushbu material eng oxirgi 7 versiyasi asosida tuzilgan.

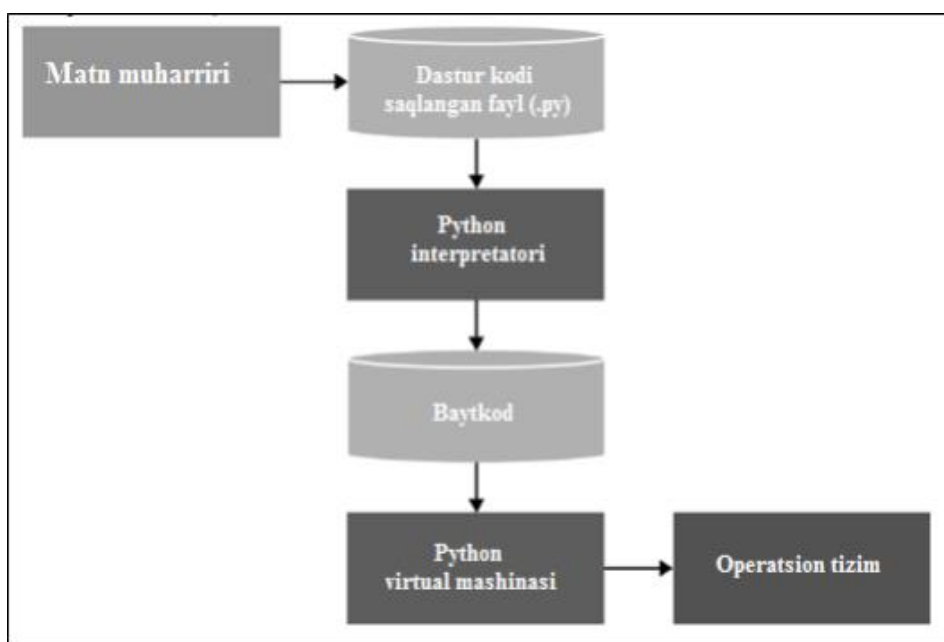
Python dasturlash tilining asosiy xususiyatlari quyidagilardan iborat: Skriptli til. Dastur kodi skriptlar ko‘rinishida bo‘ladi; Turli dasturlash paradigmlarni, xususan ob'ektga yo‘naltirilgan va funksional paradigmlarni o‘zida mujassamlagan; Skriptlar bilan ishlash uchun interpretator kerak bo‘lib, u skriptni ishga tushiradi va bajaradi.

Portativlik va platformaga bog‘liqmaslik. Kop'yuterda qanday operasion tizim - Windows, Mac OS, Linux bo‘lishidan qat'iy nazar, ushbu operasion tizimda interpretator mavjud bo‘lsa, foydalanuvchi tomonidan yozilgan skript kod bajariladi.

Xotiraning avtomatik boshqarilishi;

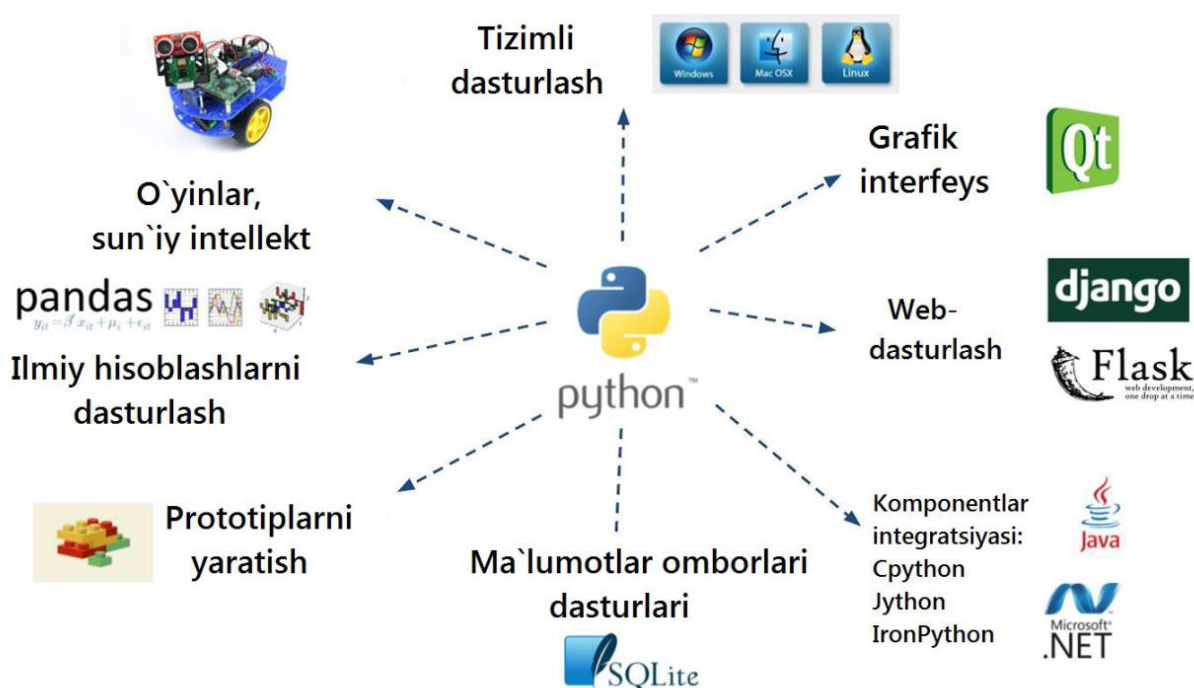
Turlarga dinamik ajratilishi;

Pythonda dasturning bajarilishi quyidagicha bo‘ladi: Dastlab mant muharririda ushbu dasturlash tili asosida ifodalar ketma-ketligidan iborat skript kod yoziladi. Ushbu yozilgan skript kod barajirilish uchun interpretatorga uzatiladi. Interpretator skript kodni oraliq baytkodga tarjima qiladi. Keyin virtual mashina baytkodni operatsion tizimda bajariladigan instruksiyalar (mashina buyruqlari) to‘plamiga o‘tkazadi.



Rasm 1. Pythonda dasturning bajarilishi jarayoni.

Shu ta'kidlash lozimki, rasman interpretator tomonidan dastlabki kodning baytkodga tarjima qilinishi va virtual mashinaning ushbu baytkodni mashina buyruqlari to'plamiga o'tkazilishi ikkita turli jarayon bo'lsada, ammo amalda ular bitta interpretatorning o'zida birlashtirilgan. Python juda oddiy dasturlash tili bo'lib, u ixcham shu bilan bir vaqtda sodda va tushinarli sintaksisga ega. Shu sababli Python o'rganish uchun juda oson til sifatida butun dunyoda eng tez tarqalayotgan tillardan biri sifatida e'tirof etiladi. Python faol qo'llaniladigan sohalarni qisqacha sanab o'tamiz:



2-rasm. Python tili sohalari

1. Tizimli dasturlash.
2. Grafik interfeysli dasturlarni ishlab Chiqarish.
3. Dinamik veb-saytlarni ishlab Chiqarish.
4. Komponentlarning integratsiyasi.
5. Ma'lumotlar bazalari bilan ishlash dasturlarini ishlab Chiqarish.
6. Tez prototiplash.
7. Ilmiy hisoblash uchun dasturlarni ishlab Chiqarish.
8. O'yinni ishlab Chiqarish.

Bundan tashqari, ushbu tilda hozirgi kunga kelib, turli sohalarga (veb, o'yin, mul'timediya, ilmiy tadqiqot) mo'ljallangan katta hajmdagi kutbxonalar majmui yaratilgan bo'lib, uning tobora mashhurlashib borishiga sabab bo'lmoqda.

Pythonni o'rnatish. Pythonda dastur tuzish uchun interpretator kerak bo'ladi. Uni kompyuteringizdagi o'rnatilgan operasion tizim turiga mos ravishda <https://www.python.org> rasmiy saytidan kerakli versiyasini tushirib olishingiz mumkin.

Pythonda dastur kodini yozish

Python tilida dastur instruktsiyalar to'plamidan tashkil topgan bo'lib, har bir instruktsiya alohida qatorda joylashgan bo'lishi kerak bo'ladi. Masalan:

```
1 print(3 + 5)
2 print("Python - dasturlash tili!")
```

Python da xat boshi (otstup) juda muhim ahamiyatga ega hisoblanadi. xat boshining noto'g'ri joylashtirilishi dasturda xatolikka olib keladi. Masalan yuqoridagi dastur kodini quyidagicha yozamiz:

```
1 print(3 + 5)
2     print("Python - dasturlash tili!")
```

Ushbu dastur kodi yuqoridagisi bilan bir xil bo'lishiga qaramasdan interpretator xatolik haqida xabar chiqaradi va dastur bajarilmaydi. Shuning uchun ham Pythonda har bir instruktsiya alohida qatorda yozilishi shart. Ushbu hususiyat Pythonning boshqa tillardan, masalan, Java, C# tillaridan farqli jihatlaridan biri hisoblanadi.

O'zgaruvchilar va berilganlar turlari

Pythonda, boshqa dasturlash tillaridagi kabi o'zgaruvchilar aniq bir turdagi berilganlarni saqlash uchun xizmat qiladi. Pythonda o'zgaruvchilar alfavit belgilari

yoki tag chizig'i belgisi bilan boshlanishi va tarkibi son, alfavit belgilari, tag chizig'i belgilaridan iborat bo'lishi, ya'ni bir so'z bilan aytganda identifikator

bo'lishi kerak. Bundan tashqari o'zgaruvchi nomi Pythonda ishlatiladigan kalit so'zlar nomi bilan mos tushmasligi shart. Masalan, o'zgaruvchi nomi and, as, assert, break, class, continue, def, del, elif, else, except, False, finally, for, from, global, if, import, in, is, lambda, None, nonlocal, not, or, pass, raise, return, True, try, while, with, yield kabi kalit so'zlar nomi bilan mos tushishi mumkin emas.

Masalan, o'zgaruvchini aniqlash (hosil qilish) quyidagicha amalga oshiriladi:

```
1 a = 14
2 name = "SDY"
```

Yuqorida a va name o'zgaruvchilari yaratildi va ularga qiymat berildi. Shuni alohida ta'kidlash kerakki, Pythonda o'zgaruvchini dastlab e'lon qilish degan tushuncha mavjud emas (masalan: c++ tilida int a kabi o'zgaruvchi e'lon qilinadi), balki o'zgaruvchi kiritiladi va unga qiymat beriladi (masalan: a=14). Berilgan qiymatga ko'ra interpretator o'zgaruvchining turini aniqlaydi. Pythonda o'zgaruvchilarni nomlashning ikki turi: "camel case" va "underscore notation" turlaridan foydalanish tavsiya qilingan.

"camel case" turida o'zgaruvchiga nom berilganda, agar o'zgaruvchi nomi alohida so'zlar birikmasidan tashkil topgan bo'lsa, ikkinchi so'zdan boshlab har bir so'zning birinchi harfi katta harfda (katta registr) bo'lishi talab qilinadi. Masalan:

```
1 firstName = "Saidov"
```

"underscore notation" turida esa so'zlar orasiga tag chizig'i "_" belgisi qo'yiladi. Masalan:

```
1 first_name = "Saidov"
```

O'zgaruvchilar biror bir turdagi berilganlarni saqlaydi. Pythonda bir necha xildagi berilganlar turlari mavjud bo'lib, ular odatda to'rtta guruhga ajratiladi:

sonlar, ketma-ketliklar, lug'atlar va to'plamlar:

bool (boolean) – True va False mantiqiy qiymatlar uchun;

int – butun sonlar uchun, butun turdagi songa kompyuter xotirasida 4 bayt joy ajratiladi;

float – suzuvchan nuqtali sonlar (haqiqiy sonlar) uchun, haqiqiy sonlarni saqlash uchun kompyuter xotirasidan 8 bayt joy ajratiladi;

complex – kompleks sonlar uchun;

str – satrlar uchun, Python x versiyasidan boshlab satrlar bu- Unicode kodirovkasidagi belgilar ketma-ketligini ifodalaydi;

bytes – 0-255 diapazondagi sonlar ketma ketligi uchun

byte array – baytlar massivi uchun;

list – ro‘yxatlar uchun;

tuple – kortejlar uchun;

set – tartiblanmagan unikal ob`ektlar kolleksiyasi uchun;

frozen set – set singari, faqat u o‘zgartirilishi mumkin emas (immutable);

dict – lug‘atlar uchun. Har bir element kalit so‘z va qiymat juftligi ko‘rinishida ifodalaniladi.

Tarmoqlanuvchi algoritmlar, if va else shart operatrlari bilan ishlash misollari. Chala tarmoqli va to‘liq tarmoqli algoritmlar.

if shart amali shart ifodalarda qo‘llanilib, uning natijasiga ko‘ra dastur bajarilishi u yoki bu yo‘lga yo‘naltiriladi. U quyidagi umumiy ko‘rinishga ega: if mantiqiy ifoda:

ifodalar

[elif mantiqiy ifoda:

ifodalar]

[else:

ifodalar]

if shart operatorining eng sodda ko‘rinishida if kalit so‘zidan keyin mantiqiy ifoda yoziladi va ikki nuqta (:) qo‘yiladi. Keyingi qatordan amallar yoziladi. Harbir amal alohida qatorda yozilishi yoki ularni nuqta vergul (;) bilan ajratgan holda bitta qatordan yozish talab qilinadi. Shuni alohida ta`kidlash kerakki Pythonda boshqa tillardagi kabi if shart amalini tana qismini ifodalovchi maxsus belgilar mavjud emas (manasal c++, c# da {,} blok belgilari ishlatiladi). Shu sababli uning tana qismidagi

ifodalar if kalit soʻziga nisbatan bitta xat boshi (toʻrtta probel belgisi) belgisi tashlab yoziladi.

Masalan:

```
1 yoshi = 21
2 if yoshi >18:
3     print("Kirishga ruxsat beriladi")
4     print("Tamom")
```

Bu erda if kalit soʻzidan keyin `yoshi >18` mantiqiy ifoda kelgan. Tana qismi bitta ifodadan tashkil topgan, yaʼni `print("Kirishga ruxsat beriladi")` va u if ga nisbatan bitta xat boshi tashlab yozilgan. Keyingi qatordagi `print("Tamom")` ifodasi if ning tana qismiga tegishli emas, shuning uchun u if bilan bir ustunda yozishgan va bu xabar shart bajarilish-bajarilmasligidan qatʼiy nazar har doim konsol ekraniga chiqariladi.

Agarda `print("Tamom")` ifodasi oldiga bitta xat boshi qoʻysak, u holda ushbu ifoda ham if blokiga tegishli boʻlib qoladi, yaʼni

```
1 yoshi = 21
2 if yoshi >18:
3     print("Kirishga ruxsat beriladi")
4     print("Tamom")
```

Ushbu holatda shart bajarilsa, ikkala xabar ham konsol ekraniga chiqariladi, aks holda hech biri chiqarilmaydi.

if shart ifodasi `false` qiymat qaytaradigan holatda qandaydir amal bajarilishini aniqlash uchun `else` blokida bajarilishi kerak boʻlgan amallar yoziladi.

Masalan:

```
1 yoshi = 21
2 if yoshi >18:
3     print("Kirishga ruxsat beriladi")
4 else:
5     print("Kirishga ruxsat berilmaydi")
```

Agar `yoshi >18` shart bajarilsa if blokidagi aks holda `else` blokidagi amallar

bajariladi.

Bir necha alternativ shartlarni ishlatish uchun qo‘shimcha elif blokidan foydalaniladi.

```
1 # ax^2+bx+c=0 kvadrat tenglama echimlari soni
2 a= int(input("a="))
3 b= int(input("b="))
4 c= int(input("c="))
5 d = b**2 - 4*a*c
6 if d >0:
7     print("Tenglama 2 ta haqiqiy echimga ega")
8 elif d == 0:
9     print("Tenglama 1 ta haqiqiy echimga ega")
10 else:
11     print("Tenglama haqiqiy echimga ega emas")
```

Mantiqiy amallar

Murakkab shartli ifodalarni yozish, odatda mantiqiy amallar yordamida amalga oshiriladi. Pythonda quyidagi mantiqiy operatorlar mavjud:

and (mantiqiy ko‘paytirish). Murakkab ifodadagi biror bir qism ifodaniqiymati False bo‘lsa, ifodaning yakuniy qiymati False , aks holda True qiymat

qaytaradi. Masalan:

```
1 yoshi = 21
2 vazni = 72
3 natija = yoshi > 17 and vazni == 72
4 print(natija) # True
```

Yuqoridagi dasturda murakkab mantiqiy amal ikki qismdan yoshi > 17 va vazni > 56 qismlardan tashkil topgan bo‘lib, ular and mantiqiy operatori bilan birlashtirilgan. Agarda ikkala mantiqiy amal True qiymat qaytarsa ifodaning qiymati True bo‘ladi, aks holda False qiymat qaytaradi.

Matriqiy ifodalarda faqatgina taqqoslash amallaridan foydalanish shart emas.

Ixtiyoriy mantiqiy amal yoki boolean turidagi qiymatlar (True, False) ham ishlatilishi mumkin. Masalan:

```
1 yoshi = 21
2 vazni = 72
3 t = True
4 natija = yoshi > 17 and vazni > 56 and t
5 print(natija) # True
```

or (mantiqiy qo‘shish). Agarda ifodadagi biror bir qism ifoda True qiymat qaytarsa, yakuniy natija ham True, aks holda False bo‘ladi.

```
1 yoshi = 21
2 t = False
3 natija = yoshi > 17 or t
4 print(natija) # True
```

not (mantiqiy inkor). Ifodaning qiymatini True bo‘lsa, natija False va aksincha.

```
1 yoshi = 21
2 t = False
3 print(not yoshi > 17) # False
4 print(not t) # True
```

and operatorining biror bir operandi False qiymatga ega bo‘lsa, u holda bshqa operand qiymati tekshirib (hisoblanib) o‘tirilmaydi, har doim natija False bo‘ladi. Bunday xususiyat ish unumdorligini bir oz bo‘lsada oshirish imkonini beradi. Xuddi shunaqa xususiyat or operatori uchun ham o‘rinli. Ya`ni or operatorining biror bir operandi qiymati True qiymatga ega bo‘lsa, boshqa operandlar tekshirilmaydi, natija sifatida har doim True qiymati qaytariladi. Agar bitta ifodada bir nechta mantiqiy operatorlar qatnashgan bo‘lsa, u holda ularning ustunligiga (prioritetiga) alohida e`tibor qatarish kerak. Dastlab not operatori keyin and va eng so‘ngra or operatori bajariladi.

Masalan:

```
1 yoshi = 22
2 xolati = False
3 vazni = 58
4 natija = vazni == 58 or xolati and not yoshi > 21 # True
5 print(natija)
```

Ushbu dasturda keltirilgan ifodadagi mantiqiy amallar quyidagi ketma-ketlikda bajariladi:

1. `not yoshi > 21` mantiqiy ifoda `False` qiymat qaytaradi;
2. `xolati and False` (`not yoshi > 21`) esa `False` qiymat qaytaradi;
`vazni == 58 or True` (`xolati and not yoshi > 21`) esa `True` qiymat qaytaradi.

Shuni alohida ta'kidlash kerarki, mantiqiy ifodalarda mantiqiy amallarning bajarilish ketma-ketligini qavslar (,) yordamida o'zgartirish mumkin.

Sikllar bilan ishlash: sikllar va siklik operatorlar. Sikllar yordamida summa va ko'paytmalarni topish

Odatda sikl operatorlari biror - bir jarayonni qandaydir shart asosida takrorlash uchun ishlatiladi. Python da sikl operatorlarining ikki turi, `while` va `for` qaraladi.

while takrorlash operatori

`while` takrorlash operatori quyidagi umumiy ko'rinishga ega:

while shart ifodasi:

instruksiyalar

`while` kalit so'zidan keyin shart ifodasi ko'rsatiladi va ushbu shart ifodasi `True` qiymat (`True`) bo'lar ekan amallar ketma-ketligi takror va takror bajarilishda davom ettiriladi. `while` operatorining barcha instruksiyalari undan keyingi qarorda

yoziladi va u while kalit soʻzidan bitta xat boshi tashlab yoziladi. Masalan:

```
1 sum = 0
2 n = int(input("n="))
3 i = 1
4 while i <= n:
5     sum = sum + i
6     i += 1
7 print("summa (1+...+n) =", sum)
```

Yuqoridagi misolda 1 dan n gacha boʻlgan sonlar yigʻindisi hisoblash dasturi while operatori yordamida amalga oshirilgan. Eʼtibor berilsa while operatorining insturktsiyalari undan keyingi qatorda bitta xat boshi tashlab yozilgan. Ushbu holatda while operatori 2 ta instuktsiyalardan tashkil topgan (sum = sum + i va i += 1).

Nazorat savollari

1. Kodlarni tuzish nima va uning asosiy komponentlari nimalardan iborat?
2. Python dasturlash tilida proyektini saqlash va qayta davom ettirish qanday amalga oshiriladi?
3. Python tilining asosiy sintaksis qoidalari qanday?
4. Tarmoqlanuvchi algoritmlar qanday ishlaydi va if-else shart operatorlari qanday qoʻllaniladi?
5. Chala tarmoqli va toʻliq tarmoqli algoritmlar oʻrtasidagi farqlar nimada?
6. Munosabatlar operatorlari qanday ishlaydi va ularning Python dasturlash tilida qoʻllanilishi qanday?
7. Tanlash va solishtirish operatorlari qanday vazifalarni bajaradi?
8. Python dasturlash tilida mantiqiy operatsiyalar qanday ishlaydi?
9. Sikllar qanday ishlaydi va ularning asosiy operatorlari nimalardan iborat?
10. Sikllardan foydalanib summa va koʻpaytma hisoblash qanday amalga oshiriladi?

Foydalanilgan adabiyotlar

1. Schmidt W., Völschow M. // Numerical Python in Astronomy and Astrophysics. A Practical Guide to Astrophysical Problem Solving, Springer, 2021. — 257 p. — ISBN 978-3-030-70346-2
2. T.Axmadjonov, M. Akramov // Fizik jarayonlarni kompyuterda modellashtirish // T.: “Universitet”, 2024
3. M.Akramov, T.Axmadjanov // Fizik jarayonlarni kompyuterda modellashtirish, o‘ quv qo‘llanma, T., 2024
4. Нагаева И.А., Кузнецов И.А. // Основы математического моделирования и численные методы, учебное пособие, Лань.:, М. 2024, 204 с.
5. Слабнов В. Д. // Численные методы, учебное пособие, Лань.:, М. 2024, 392 с.
6. Грабовский, Р.И. Курс физики: Учебник / Р.И. Грабовский. – СПб.: Лань, 2012. – 608 с.
7. Фаритов, Т.А. Курс общей физики: Учебник / Т.А. Фаритов. – СПб.: Лан П, 2016. – 656 с.

Mavzu 3. Massivlar: bir o‘lchamli va ikki o‘lchamli massivlarni shakillantirish, elementlari bilan islah. Tasodifiy jarayonlar haqida tushuncha va tasodifiy sonlar bilan islah. Grafiklar chizish usullari

Reja:

1. Massivlar: bir o‘lchamli va ikki o‘lchamli massivlarni shakillantirish, elementlari bilan islah.
2. Tasodifiy jarayonlar haqida tushuncha va tasodifiy sonlar bilan islah.
3. Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.

***Tayanch iboralar:** Massivlar, bir o‘lchamli va ikki o‘lchamli massivlarni shakillantirish, elementlari bilan islah, tasodifiy jarayonlar, tasodifiy sonlar ,*

Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.

Python dasturlash tilida kompyuter xotirasiga bir o'zgaruvchi yordamida bir nechta qiymatlarda foydalanish bo'yicha bir nechta turlarni qarab o'tdik. Bir o'zgaruvchi bilan bir nechta qiymatlarni ro'yxat, kortej, to'plam va satr turlar bilan amalga oshirish mumkin. Python dasturlash tilida bir o'zgaruvchi yordamida bir nechta qiymatlardan foydalanish uchun massiv degan turdan ham foydalanish imkoniyati mavjud. Bu qismda dasturdagi ma'lumot strukturalari bilan tanishishni boshlaymiz. Dasturda ikki asosiy tur ma'lumot strukturalari mavjuddir. Birinchisi statik, ikkinchisi dinamikdir. Statik deganimizda xotirada egallagan joyi o'zgarmas, dastur boshida beriladigan strukturalarni nazarda tutamiz.

Dinamik ma'lumot tiplari dastur davomida o'z hajmini, egallagan xotirasini o'zgartirishi mumkin. Massivlar hotirada ketma-ket joylashgan, ma'lumotlar guruhidir. Alohida bir o'zgaruvchini ko'rsatish uchun massiv nomi va kerakli o_zgaruvchi indeksini yoziladi.

Bir turga mansub bo'lgan yagona nom bilan saqlanuvchi tartiblangan ma'lumotlar majmuasi massiv deyiladi. Massivlar yagona o'zgaruvchi bilan kompyuter xotirasiga saqlanadi, uning elementlari ma'lum bir indekslar bilan tartiblab joylashtiriladi. Bu qonuniyat ro'yxatlar uchun ham o'rinli bo'ladi. Massivlar yagona nom bilan bir nechta qiymatni o'zida mujassamlashtiradi, bularga matematikadagi vektorlarni misol keltirish mumkin. Vektor ham yagona nom bilan saqlanib uning tarkibida bir nechta qiymatni o'zida mujassamlashadi. Vektorning ham elementlari bir turga mansub va tartiblangan bo'ladi.

Pythondagi massiv - bu bir xil turdagi ob'ektlarni saqlash uchun ishlatiladigan buyurtma qilingan ma'lumotlar tuzilishi. Funktsional imkoniyatlari jihatidan ular ro'yxatlarga o'xshashdir, ammo ularning kirish ma'lumotlari turiga, shuningdek o'lchamlariga nisbatan ba'zi cheklovlar mavjud. Ushbu xususiyatga qaramay, massivlar Python dasturlash tilidagi ma'lumotlar to'plamlari bilan ishlash uchun juda funktsional vosita hisoblanadi.

Massivlarni yaratish va to'ldirish.

Pythonda yangi qator qo'shishdan (yaratishdan) oldin, bunday ob'ekt bilan ishlash uchun mas'ul bo'lgan kutubxonani import qilishingiz kerak. Buning uchun dastur fayliga `from array import *` qator qo'shilishi kerak. Massivlar bitta doimiy ma'lumotlar turi bilan o'zaro aloqada bo'lishga qaratilgan bo'lib, natijada ularning barcha elementlari bir xil o'lchamga ega. array funksiyasidan foydalanib biz yangi ma'lumotlar to'plamini yaratishimiz mumkin.

Massivlarni yaratishning umumiy sintaksisi quyidagicha:

```
array(massiv_turi, qiymatlar_ro'yxati)
```

Quyidagi misol Python massivni qanday to'ldirish kerakligini ko'rsatib beradi: `from array import *`

```
massiv = array('i', [2, 5, 4, 0, 8])
```

Massiv funksiyasi ikkita argumentni oladi, birinchisi - bu yaratilgan massivning turi, ikkinchisi - uning qiymatlarining dastlabki ro'yxati. Bu yerda massiv elementlarining 'i' (2 baytli butun) tur. Buning o'rniga 1 baytli belgi 'c' (char turi)ni yoki 4 baytli 'f' float turini kabi boshqa turlardan foydalanishimiz mumkin. Quyidagi jadvalda massiv turlari keltirilgan:

4.4.1-jadval

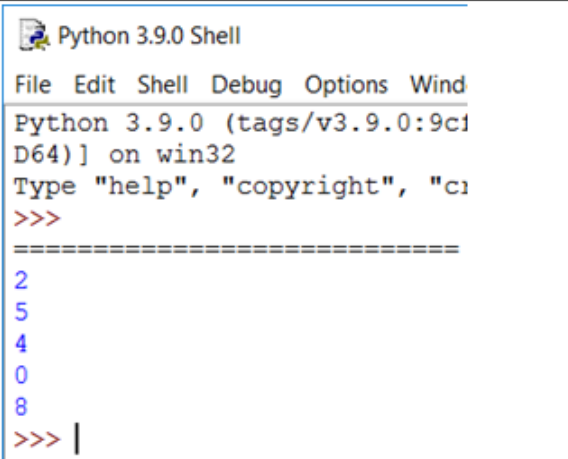
Turning massivda yozilishi	C turi	Python turi	Minimal hajmi baytda
'b'	signed char	int	1
'B'	unsigned char	int	1
'u'	Py_UNICODE	unicode character	2
'h'	signed short	int	2
'H'	unsigned short	int	2
'i'	signed int	int	2

'T'	unsigned int	int	2
't'	signed long	int	4
'L'	unsigned long	int	4
'q'	signed long long	int	8
'Q'	unsigned long long	int	8
'f'	Float	float	4
'd'	Double	float	8

Shuni esda tutish kerakki, massiv faqat bitta turdagi ma'lumotlarni saqlashi mumkin, aks holda dasturni ishga tushirganimizda xatolik beradi va muvaffaqiyatsiz bo'ladi.

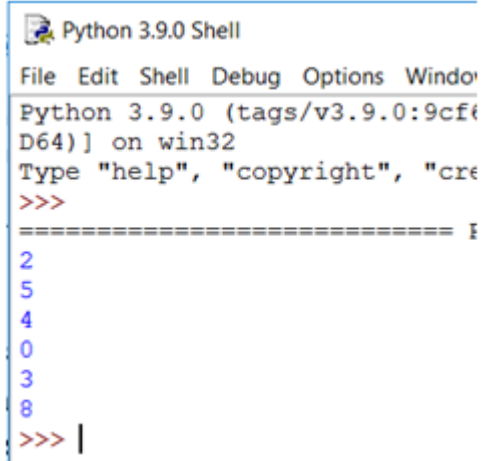
Massiv elementiga murojaat qilish. Kvadrat qavs yordamida massiv elementiga murojaat qilishimiz mumkin. Masalan : massiv[2].

Massivlarni ekranga chiqarish. Dasturdagi har qanday ma'lumotlar bilan ishlashda vaqti-vaqti bilan ularni tekshirishga ehtiyoj bor. Bu ularni ekranda aks ettirish orqali osonlikcha amalga oshiriladi. Buni amalga oshirish uchun print deb nomlangan funktsiya yordam beradi. Bu ilgari yaratilgan va to'ldirilgan qator elementlaridan birini argument sifatida qabul qiladi. Quyidagi misolda for sikl operatori yordamida ma'lumotlar massivining har bir elementi vaqtinchalik identifikator i orqali chiqariladi:

<pre>from array import * massiv = array('i', [2, 5, 4, 0, 8]) for i in massiv: print(i)</pre>	 <pre>Python 3.9.0 Shell File Edit Shell Debug Options Wind Python 3.9.0 (tags/v3.9.0:9c1 D64) on win32 Type "help", "copyright", "c >>> ===== 2 5 4 0 8 >>> </pre>
---	--

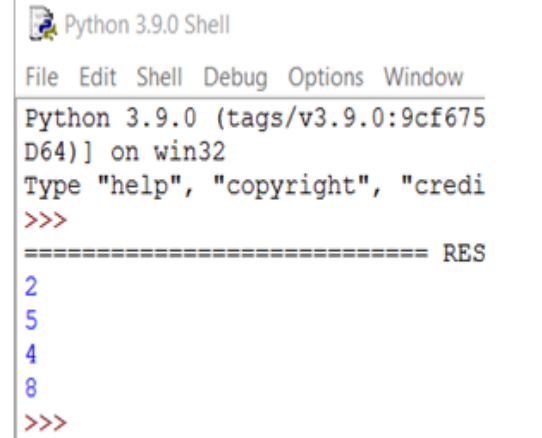
Yuqoridagi kodning natijasida barcha element qiymatlari bo‘yicha takrorlanadi va ekranga chiqariladi.

Massivga element qo‘shish. Python qatoriga yangi element qo‘shish uchun insert metodidan foydalanish kerak. Buning uchun uni avval yaratilgan ob’ekt orqali chaqirish va ikkita qiymatni argument sifatida kiritish kerak. Birinchisi (4) massivdagi yangi elementning indeksiga, ya’ni uni joylashtirish kerak bo‘lgan joyga, ikkinchisi (3) qiymatning o‘zi uchun javobgardir.

<pre>from array import * massiv = array('i', [2, 5, 4, 0, 8]) massiv.insert(4, 3) for i in massiv: print(i)</pre>	 <pre>Python 3.9.0 Shell File Edit Shell Debug Options Windo Python 3.9.0 (tags/v3.9.0:9cfe D64) on win32 Type "help", "copyright", "cre >>> ===== 2 5 4 0 3 8 >>> </pre>
---	--

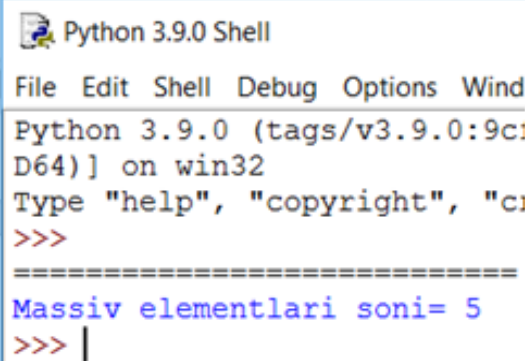
Shuni esda tutish kerakki, biz qatorga faqat ilgari yaratilgan ob’ekt tegishli bo‘lgan turdagi ma’lumotlarni qo‘shishimiz mumkin. Bunday operatsiyani bajarishda mavjud bo‘lgan elementlar soni dasturning ehtiyojlariga qarab ko‘payadi.

Elementni o‘chirish. Pythonda pop() metodi yordamida keraksiz elementlarni massivdan olib tashlash mumkin, uning argumenti yacheka indeksi (3). Yangi element qo‘shilgandek bo‘lgani kabi, usulni misolda ko‘rsatilgandek, avval yaratilgan ob’ekt orqali chaqirish kerak.

<pre> from array import * massiv = array('i', [2, 5, 4, 0, 8]) massiv.pop(3) for i in massiv: print(i) </pre>	 <pre> Python 3.9.0 Shell File Edit Shell Debug Options Window Python 3.9.0 (tags/v3.9.0:9cf675 D64) on win32 Type "help", "copyright", "credi >>> ===== RES 2 5 4 8 >>> </pre>
---	---

Ushbu operatsiyani bajargandan so‘ng, mavjud bo‘lgan xotira katakchalari soni elementlarning joriy soniga to‘g‘ri keladigan qilib massiv tarkibini o‘zgartiradi.

Massiv uzunligini olish. Dastur bajarilishida massivning uzunligi o‘zgarishi mumkinligi sababli, ba’zida uning tarkibidagi elementlarning hozirgi sonini bilish foydalidir. len() metodi Pythondagi massivning uzunligini (hajmini) butun son sifatida olish uchun ishlatiladi. Pythonda massiv elementlari sonini ekranga chiqarish uchun print() metodidan foydalanamiz:

<pre> from array import * massiv = array('i', [2, 5, 4, 0, 8]) print("Massiv elementlar isoni=",len(massiv)) </pre>	 <pre> Python 3.9.0 Shell File Edit Shell Debug Options Wind Python 3.9.0 (tags/v3.9.0:9cf675 D64) on win32 Type "help", "copyright", "credi >>> ===== Massiv elementlari soni= 5 >>> </pre>
---	--

Yuqoridagi dastur kodidan ko‘rinib turibdiki, print() metodi argumenti sifatida len natijasini oladi, bu esa konsolga raqamli qiymatni chiqarishga imkon beradi.

Pythonda ikki va ko‘p o‘lchovli massivlar

Massivlar holatiga ko‘ra ikki turga bo‘linadi.

- Bir o‘lchovli massivlar;
- Ikki o‘lchovli massivlar;

Bir o‘lchovli massivlar ma‘lumotlarni bir satrli ko‘rinishda saqlansa , ikki o‘lchovli massivlar esa ma‘lumotlarni satrlar satri ko‘rinishida saqlaydi. Python

dasturlash tilida massivlarni qayta ishlash uchun numpy kutubxonasini faollashtirish kerak Dastur tuzishdan oldin albatta numpy numpy kutubxonasini faollashtirish kerak.

Bir o'lchovli massivlar

Odatda massivlar zarurat, katta hajmdagi tartiblangan, lekin chekli elementlarga oid masalalarni hal etishda yuzaga keladi. Massivning ixtiyoriy elementiga murojat qilish uchun uning indeks nomeri bo'yicha murojat qilinadi. Bir o'lchovli massivlarni python dasturlash tilida faollashtirish mumkin.

Bir o'lchovli massivlarni boshlang'ich qiymatlari bergan holda python dasturlash tilida quyidagicha faollashtiriladi.

```
<massiv o'zgaruvchisi>=array([x1,x2,...])
```

Massivni Python dasturlash tilida faollashtirish uchun, elementlar soni yoki massiv elementlarining boshlang'ich qiymatlari berilishi shart emas. Python dasturlash tilida bir o'lchovli massivni faollashtirish quyidagicha amalga oshiriladi. Massivni umumiy ko'rinishida birinchi massiv o'zgaruvchisi va massiv elementlari yoziladi. Massiv elementlari, ixtiyoriy turdagi sondan iborat bo'lishi mumkin. Lekin massiv elementlari tarkibida bitta haqiqiy son bo'lsa boshqa elementlari ham haqiqiy deb qaraladi.

Massiv elementlari soni biror bir ifoda yoki yagona o'zgaruvchi bo'lishi mumkin, bitta o'zgaruvchi orqali massivning umumiy indekslarini ifodalash mumkin. Massiv elementlarini ustida amallar bajarishni quyidagi masala orqali qaraymiz. Massiv elementlarini tartib nomeri doimo 0 dan boshlanadi. Python dasturlash tilida massiv elementlari boshlang'ich qiymatlarini bermasdan faollashtirishh quyidagicha.

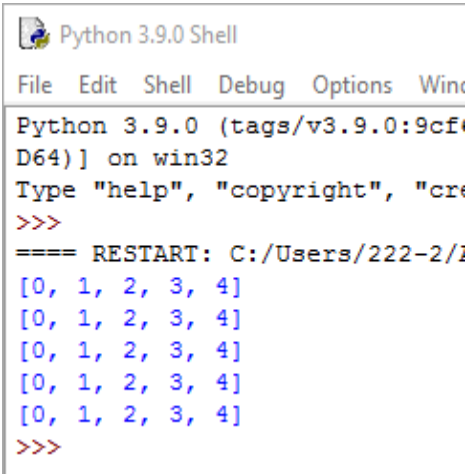
```
<massiv o'zgaruvchisi>=array([])
```

Bunda massiv elementlari kvadrat qavs ichida yozilmaydi ya'ni bo'sh holda qoldiriladi, bunda massiv elementsiz shakllanadi. Massiv elementlar soni uning tarkibidagi qiymatlar orqali aniqlanadi va massivning elementlari doimo nolinchidan boshlanadi. Massiv elementlarini funksiyalar orqali va klaviatura yordamida hosil qilish Massiv elementlarini python dasturlash tilida ma'lum bir

ketma ketlik yoki funksiya orqali hosil qilish imkoniyati mavjud. Bunda massiv elementlari berilgan funksiyaning qiymatlarini qabul qiladi. Range() funksiyasi orqali massiv elementlarini shakllantirishni quyidagi dastur orqali hal etamiz.

Ikki o'lchovli massiv.

Ba'zi hollarda oddiy bir o'lchovli massiv ma'lum bir ma'lumot to'plamini to'g'ri ko'rsatish uchun etarli emas. Python dasturlash tilida ikki o'lchovli va ko'p o'lchovli massivlar mavjud emas, ammo ushbu platformaning asosiy imkoniyatlari ikki o'lchovli ro'yxatni tuzishni osonlashtiradi. Ushbu dizayn elementlari quyidagi misolda ko'rsatilgandek to'ldirilib, ustunlar va qatorlarga joylashtirilgan.

<pre>from array import *d1 = [] for j in range(5): d2 = [] for i in range(5): d2.append (i) d1.append(d2)for i in d1: print(i)</pre>	
---	---

Nazorat savollari

- 1. Python dasturlash tilida bir o'lchamli massivlarni qanday shakllantirish mumkin?*
- 2. Ikki o'lchamli massivlarni yaratish va ularning elementlari bilan ishlash qanday amalga oshiriladi?*
- 3. Tasodifiy jarayonlar haqida tushuncha bering va Python'da tasodifiy sonlar bilan ishlash qanday amalga oshiriladi?*
- 4. random kutubxonasi yordamida tasodifiy sonlar qanday hosil qilinadi?*
- 5. Python'da matplotlib va pyplot kutubxonalarining asosiy imkoniyatlari qanday?*
- 6. Pyplot kutubxonasida chiziqli grafikni qanday chizish mumkin?*
- 7. Histogramma chizish uchun qanday funktsiyalardan foydalanish mumkin?*
- 8. Scatter plot (nuqtali grafik) yaratish uchun qanday parametrlar ishlatiladi?*

9. *Matplotlib* 'da x va y o'qlari uchun etiketlarni qanday qo'shish mumkin?
10. Turli shakildagi grafiklarni chizishda *pyplot* kutubxonasining qanday xususiyatlari mavjud?

Foydalanilgan adabiyotlar

1. Schmidt W., Völschow M. // Numerical Python in Astronomy and Astrophysics. A Practical Guide to Astrophysical Problem Solving, Springer, 2021. — 257 p. — ISBN 978-3-030-70346-2
2. T.Axmadjonov, M. Akramov // Fizik jarayonlarni kompyuterda modellashtirish // T.: "Universitet", 2024
3. M.Akramov, T.Axmadjanov // Fizik jarayonlarni kompyuterda modellashtirish, o'quv qo'llanma, T., 2024
4. Нагаева И.А., Кузнецов И.А. // Основы математического моделирования и численные методы, учебное пособие, Лань.:, М. 2024, 204 с.
5. Слабнов В. Д. // Численные методы, учебное пособие, Лань.:, М. 2024, 392 с.
6. Грабовский, Р.И. Курс физики: Учебник / Р.И. Грабовский. – СПб.: Лань, 2012. – 608 с.
7. Фаритов, Т.А. Курс общей физики: Учебник / Т.А. Фаритов. – СПб.: Лань, 2016. – 656 с.

Mavzu 4. Eksperiment ma'lumotlarining statistik tahlili. Sonli integrallash usullari va alqoritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va alqoritmlari. Ular yordamida aniq integrallarni hisoblash. Oddiy differensial tenglamalarni sonli yechish usullari va alqoritmalri.

REJA

1. Eksperiment ma'lumotlarining statistik tahlili.
2. Sonli integrallash usullari va alqoritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va alqoritmlari.
3. Ular yordamida aniq integrallarni hisoblash.
4. Oddiy differensial tenglamalarni sonli yechish usullari va alqoritmalri..

Tayanch iboralar: Eksperiment ma'lumotlarining statistik tahlili, sonli integrallash usullari, to'g'ri burchaklar usuli, trapetsiyalar usuli, Simpson usuli, aniq integrallarni hisoblash, oddiy differensial tenglamalarni sonli yechish, sonli yechish algoritmlari.

Ma'lumotlarni tahlil qilish — [matematika](#) va informatikaning eksperimental (keng ma'noda) ma'lumotlaridan bilim olish uchun eng umumiy matematik usullar va hisoblash [algoritm](#)larini qurish va o'rganish bilan shug'ullanadigan sohasi^[1]. Foydali ma'lumotlarni olish va qarorlar qabul qilish uchun ma'lumotlarni o'rganish, filtrlash, o'zgartirish va modellashtirish jarayoni ma'lumotlarni tahlil qilishda ko'plab jihatlar va yondashuvlarga ega, fan va faoliyatning turli sohalarida turli usullarni qamrab oladi.

Ma'lumotlarni qazib olish — bu ma'lumotlarni tavsiflashdan ko'ra uni modellashtirish va kashf etishga qaratilgan ma'lumotlarni tahlil qilishning o'ziga xos usuli. Biznes razvedka yig'ishga asoslangan ma'lumotlar tahlilini qamrab oladi. [Statistik](#) ma'noda, ba'zida ma'lumotlar tahlilini tavsiflovchi statistika, kashfiyot ma'lumotlarini tahlil qilish va statistik gipotezani tekshirishga ajratadilar. Tashkilot ma'lumotlarini tahlil qilish ma'lumotlarning yangi xususiyatlarini aniqlash bilan bog'liq bo'lsa, statistik gipotezalarni tekshirish mavjud farazlarni tasdiqlash yoki rad etish bilan bog'liqdir. [Bashoratli tahlil](#) bashorat qilish yoki tasniflash uchun statistik yoki tizimli modellarni qo'llashga qaratilgan bo'lsa, matn tahlili tuzilmagan ma'lumotlarga tegishli matn manbalaridan ma'lumotlarni ajratib olish va tasniflash uchun statistik, lingvistik va tizimli usullarni qo'llaydi. Bu ma'lumotlarni tahlil qilishning barcha turlaridir.

Ma'lumotlar integratsiyasi ma'lumotlar tahlilining asosi hisoblanadi va ma'lumotlar tahlilining o'zi ma'lumotlarni vizualizatsiya qilish va ma'lumotlarni tarqatish bilan chambarchas bog'liq. "Ma'lumotlar tahlili,, atamasi ba'zan ma'lumotlarni modellashtirish uchun sinonim sifatida ishlatiladi.

Ma'lumki, matematik statistika eksperimental ma'lumotlarning aniq ekanligini baholash hamda shunday joiz chegaralarni, chiqarilgan xulosalar aniq va

etarlicha ishonchli bo'ladigan chegaralarni, belgilash imkoniyatini beradi. O'rganilayotgan kattaliklarning aloqasini, o'zora bog'lanishini, o'zora nisbatini yoki korrelyatsiyasini(munosabatdaligini) o'rnatish uchun statistik usullar muxim o'rin tutadi.

Shuningdek, statistik metodlardan(usullardan) nazariy ma'lumotlar asosida qurilgan o'zgarish tavsiflarining amalda olingan natijalar asosida qurilgan tavsifga nisbatan qancha farq qilishini ham aniq baholash imkoni paydo bo'ladi. Nazariy ma'lumotlar amaliy olingan natijalardan keskin farq qilgan holatlarda ilmiy gipoteza olib tashlanishi (tashlab yuborilishi) yoki aksincha nazariy va amaliy ma'lumotlar yaqin kelsa, gipoteza qabul qilinishi mumkin bo'ladi. Yuqoridagilarni amalga oshirishda statistik usullarni birdan-bir maqsadga aylantirish mumkin emas, balkim qishloq xo'jaligi ob'ektlarining xossalaridan kelib chiqqan holatdan tadqiqiy natijalariga to'g'ri ishlov berish va tahlil qilishda yordamchi vosita bo'lishi talab etiladi. ya'ni bu usullar tajriba o'tkazuvchining maxoratiga sira ham ta'sir ko'rsata olmaydi. Ular faqat matematik apparat bo'lib EHM bazasida ma'lumotlar usulida arifmetik amallarni bajaradi xolos.

Demak, tajriba o'tkazuvchi inson malakasiz bo'lsa, yoki tajriba noto'g'ri o'tkazilsa eng zo'r statistik ishlov berish usullari ham to'g'ri xulosalar chiqarishga ojizlik qiladi.

Tajriba o'tkazuvchining asosiy vazifasi bu sifatli va maqsadga yo'naltirilgan tajribalarni o'tkazish, matematik statistika esa agronomik tadqiqotlarni to'g'ri olib borishda tajriba o'tkazish uchun optimal sharoitlarni tanlashda yordam beradi, eksperimental ma'lumotlarga ob'ektiv, sonli baho beradi.

Matematik statistika usullari hodisalarning mohiyatini ochib bermaydi, ammo mazkur hodisalarning miqdoriy tomonini shakillantiradi. Statistik ishlov berishning tadqiqotlar olib borishdagi o'rni shu bilan yuqori baholanadiki, tajribalar o'tkazish amaliyoti shuni tasdiqlab turibdi, eksperimentlarning barcha jarayonlarida: tajriba boshlanishidan to oxirgi yakuniy xulosalar qabul qilinganicha rejali tajribalarni o'tkazish va statistik ishlov berishlar tajriba o'tkazuvchining yo'ldoshi bo'lishi kerak. Ya'ni, statistik tahlil usullaridan uzluksiz va muntazam foydalanib

borish zarur.

Sonli integrallash - bu matematik analizda aniq integrallarni taxminiy hisoblash usuli bo'lib, uzluksiz funksiyalarni diskret qiymatlar orqali yaqinlashtirish tamoyiliga asoslanadi.

Sonli integrallashning asosiy afzalliklari:

Analitik integrallash qiyin yoki imkonsiz bo'lgan funksiyalar uchun qo'llaniladi.

Yuqori aniqlikdagi hisob-kitoblar uchun moslashuvchanlikni ta'minlaydi. Real dunyo muammolarini modellashtirish va simulyatsiya qilish imkoniyatini beradi. Bu usul analitik yechim topish qiyin yoki imkonsiz bo'lgan hollarda qo'llaniladi.

To'g'ri burchaklar usuli

To'g'ri burchaklar usuli integrallashda funksiya qiymatlarini diskret nuqtalarda baholash orqali integrallash jarayonini yaqinlashtirish usulidir.

Asosiy formula:

$$I \approx \sum f(x_i) * \Delta x$$

Algoritm:

1. Funksiya oraliqni $[a, b]$ va bo'lak sonini n aniqlash.

2. Bo'lak uzunligini hisoblash: $\Delta x = (b-a) / n$

Yig'indini hisoblash: $I \approx \sum f(x_i) * \Delta x$

4. Natijani qaytarish.

Trapetsiya usuli

Trapetsiya usuli integrallash jarayonini qator trapetsiyalar yordamida yaqinlashtirishga asoslangan.

Formula:

$$I \approx (\Delta x / 2) \sum (f(x_i) + f(x_{i+1}))$$

Algoritm:

1. Funksiya oraliqni $[a, b]$ va bo'lak sonini n aniqlash.

2. Bo'lak uzunligini hisoblash: $\Delta x = (b-a) / n$

Integrallash summasini hisoblash: $I = f(a) + f(b)$

4. Yig'indi: $I = I + 2 \sum f(x_i)$
5. Yakuniy natija: $I = (\Delta x / 2) * I$
6. Natijani qaytarish.

Simpson (parabolalar) usuli

Simpson usuli kvadratik interpolatsiyaga asoslangan bo'lib, integrallashni parabolalar yordamida yaqinlashtiradi.

Formula:

$$I \approx (\Delta x / 3) \sum (f(x_i) + 4f(x_{i+1}) + f(x_{i+2}))$$

Algoritm:

1. Funksiya oraliqni $[a, b]$ va bo'lak sonini n aniqlash (n juft bo'lishi shart!).
2. Bo'lak uzunligini hisoblash: $\Delta x = (b-a) / n$

Integrallash summasini hisoblash:

- $I = f(a) + f(b)$
- $I = I + 4 \sum f(x_i)$ (toq indekslar uchun)
- $I = I + 2 \sum f(x_i)$ (juft indekslar uchun)

4. Yakuniy natija: $I = (\Delta x / 3) * I$
5. Natijani qaytarish.

Sonli integrallashning qo'llanilish sohalari

- Fizikada harakat qonuniyatlarini o'rganish (masalan, tezlik va yo'lni hisoblash).
- Differensial tenglamalarni yechishda (masalan, elektr zanjirlarida oqim o'zgarishlarini aniqlash).
- Iqtisodiyot va statistik tadqiqotlarda (masalan, taqsimot funksiyalarini hisoblash).
- Muhandislik va texnologik jarayonlarni modellashtirishda (masalan, suyuqlik dinamikasi).

Nazorat savollari

1. Eksperiment ma'lumotlarining statistik tahlili qanday amalga oshiriladi?
2. To'g'ri burchaklar usuli yordamida aniq integral qanday hisoblanadi?

3. Trapetsiya usuli bilan aniq integralni hisoblash jarayoni qanday amalga oshiriladi?
4. Simpson (parabolalar) usuli qanday ishlaydi va qachon qo'llaniladi?
5. Sonli integrallash algoritmlarining taqqoslanishi va samaradorligi qanday baholanadi?
6. Oddiy differensial tenglamalarni sonli usullar yordamida yechishning asosiy usullari nimalardan iborat?
7. Differensial tenglamalarni sonli yechishda qanday xatoliklar yuzaga kelishi mumkin?
8. Sonli integrallash nima va uning qo'llanilish sohalari qanday?
9. To'g'ri burchaklar usuli qanday ishlaydi va uning xatoliklari nimada?
10. Trapetsiya usuli qaysi tamoyilga asoslangan va qachon ishlatiladi?
11. Simpson usuli qanday ishlaydi va uning afzalliklari nimalardan iborat?
12. Sonli integrallash usullaridan qaysi biri eng aniq natijalarni beradi?
13. Sonli integrallash usullarining nisbiy xatolik darajasi qanday aniqlanadi?
14. To'g'ri burchaklar usulidan farqli ravishda trapetsiya usuli qanday yaxshilangan?
15. Differensial tenglamalarni yechishda sonli integrallash qanday qo'llaniladi?
16. Python va MATLAB'da sonli integrallash algoritmlari qanday amalga oshiriladi?
17. Amaliyotda sonli integrallash usullari qaysi muammolarni hal qilishda ishlatiladi?

Foydalanilgan adabiyotlar

1. Schmidt W., Völschow M. // Numerical Python in Astronomy and Astrophysics. A Practical Guide to Astrophysical Problem Solving, Springer, 2021. — 257 p. — ISBN 978-3-030-70346-2
2. T.Axmadjonov, M. Akramov // Fizik jarayonlarni kompyuterda modellashtirish // T.: "Universitet", 2024
3. M.Akramov, T.Axmadjanov // Fizik jarayonlarni kompyuterda modellashtirish, o'quv qo'llanma, T., 2024

4. Нагаева И.А., Кузнецов И.А. // Основы математического моделирования и численные методы, учебное пособие, Лань.:, М. 2024, 204 с.
5. Слабнов В. Д. // Численные методы, учебное пособие, Лань.:, М. 2024, 392 с.
6. Грабовский, Р.И. Курс физики: Учебник / Р.И. Грабовский. – СПб.: Лань, 2012. – 608 с.
7. Фаритов, Т.А. Курс общей физики: Учебник / Т.А. Фаритов. – СПб.: Лан П, 2016. – 656 с.

AMALIY MASHG'ULOT

MAVZU 1. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for data science. Python redaktorlarini (Spyder, PyCharm, VS Code) o'rnatish. Dasturlar tuzishning dastlabki qadamlari, chiziqli algoritmlar va oddiy misollar.

Reja

1. Dasturiy ta'minot: Spyder/MatlabWork vositalari va tillari. Ularni kompyuterda o'rnatish yo'llari: Anaconda, Python for data science.
2. Python redaktorlarini (Spyder, PyCharm, VS Code) o'rnatish. Dasturlar tuzishning dastlabki qadamlari, chiziqli algoritmlar va oddiy misollar.

Dasturiy ta'minot

Python dasturlash tilini yaratilishi 1990-yil boshlaridan boshlangan. O'sha paytlarda uncha taniqli bo'lmagan Gollandiyaning CWI institute xodimi Gvido van Rossum ABC tilini yaratilish proektida ishtirok etgan edi. ABCtili Basic tili o'rniga talabalarga asosiy dasturlash konsepsiyalarini o'rgatish uchun mo'ljallangan til edi. Bir kun Gvido bu ishlardan charchadi va 2 hafta davomida o'zining Macintoshida boshqa oddiy tilning interpretatorini yozdi, bunda u albatta ABC tilining ba'zi bir g'oyalarini o'zlashtirdi.

Shuningdek, Python 1980-1990-yillarda keng foydalanilgan Algol-68, C, C++, Modul3 ABC, SmallTalk tillarining ko'plab xususiyatlarini o'ziga olgandi. Gvido van Rossum bu tilni internet orqali tarqata boshladi. Bu paytda o'zining "Dasturlash tillarining qiyosiy taqrizi" veb sahifasi bilan internetda to 1996-yilgacha Stiv Mayevskiy ismli kishi taniqli edi. U ham Macintoshni yoqtirardi va bu narsa uni Gvido bilan yaqinlashtirdi. O'sha paytlarda Gvido BBC ning "Monti Paytonning havo sirki" komediyasining muxlisi edi va o'zi yaratgan tilni Monti Payton nomiga Python deb atadi (ilon nomiga emas). Til tezda ommalashdi. Bu dasturlash tiliga qiziqqan va tushunadigan foydalanuvchilar soni ko'paydi.

Boshida bu juda oddiy til edi. Shunchaki kichik interpretator bir nechta funksiyalarga ega edi. 1991-yil birinchi OYD(Obyektga Yo'naltirilgan Dasturlash) vositalari paydo bo'ldi. Bir qancha vaqt o'tib Gvido Gollandiyadan Amerikaga ko'chib o'tdi. Uni NRI korporatsiyasiga ishlashga taklif etishdi. U o'sha yerda ishladi va korporatsiya shug'ullanayotgan proektlarni Python tilida yozdi va bo'sh ish vaqtlarida tilni interpretatorini rivojlantirib bordi. Bu 1990-yil Python 1.5.2 versiyasi paydo bo'lguncha davom etdi. Gvidoning asosiy vaqti korporatsiyani proektlarini yaratishga ketardi bu esa unga yoqmasdi. Chunki uning Python dasturlash tilini rivojlantirishga vaqti qolmayotgandi. Shunda u o'ziga tilni rivojlantirishga imkoniyat yaratib bera oladigan homiy izladi va uni o'sha paytlarda endi tashkil etilgan BeOpen firmasi qo'llab quvvatladi. U CNRI dan ketdi, lekin shartnomaga 8 binoan u Python 1.6 versiyasini chiqarib berishga majbur edi. BeOpen da esa u Python 2.0 versiyani chiqardi. 2.0 versiyasi bu oldinga qo'yilgan katta qadamlardan edi. Bu versiyada eng asosiysi til va interpretatorni rivojlanish jarayoni ochiq ravishda bo'ldi.

Shunday qilib 1.0 versiyasi 1994-yil chiqarilgan bo'lsa, 2.0 versiyasi 2000-yil, 0 versiyasi esa 2008-yil ishlab chiqarildi. Hozirgi vaqtda uchinchi versiyasi keng qo'llaniladi.

Python dasturlash tili imkoniyatlari Python – bu o'rganishga oson va shu bilan birga imkoniyatlari yuqori bo'lgan oz sonlik zamonaviy dasturlash tillari qatoriga kiradi. Python yuqori darajadagi ma'lumotlar strukturasi va oddiy lekin samarador obyektga yo'naltirilgan dasturlash uslublarini taqdim etadi.

Pythonning o'ziga xosligi

1. Oddiy, o'rganishga oson, sodda sintaksisga ega, dasturlashni boshlash uchun qulay, erkin va ochiq kodlik dasturiy ta'minot.
2. Dasturni yozish davomida quyi darajadagi detallarni, misol uchun xotiraniboshqarishni hisobga olish shart emas.
3. Ko'plab platformalarda hech qanday o'zgartirishlarsiz ishlay oladi.
4. Interpretatsiya qilinadigan til.

5. Kengayishga moyil til. Agar dasturni biror joyini tezroq ishlashini xoxlasak shu qismni C yoki C++ dasturlash tillarida yozib keyin shu qismni python kodi orqali ishga tushirsa(chaqirsa) bo'ladi.

6. Juda ham ko'p xilma-xil kutubxonalarga ega.

7. xml/html fayllar bilan ishlash

8. http so'rovlari bilan ishlash

9. GUI(grafik interfeys)

10. Veb saytlarni yaratish

11. FTP bilan ishlash

12. Rasmi audio video fayllar bilan ishlash

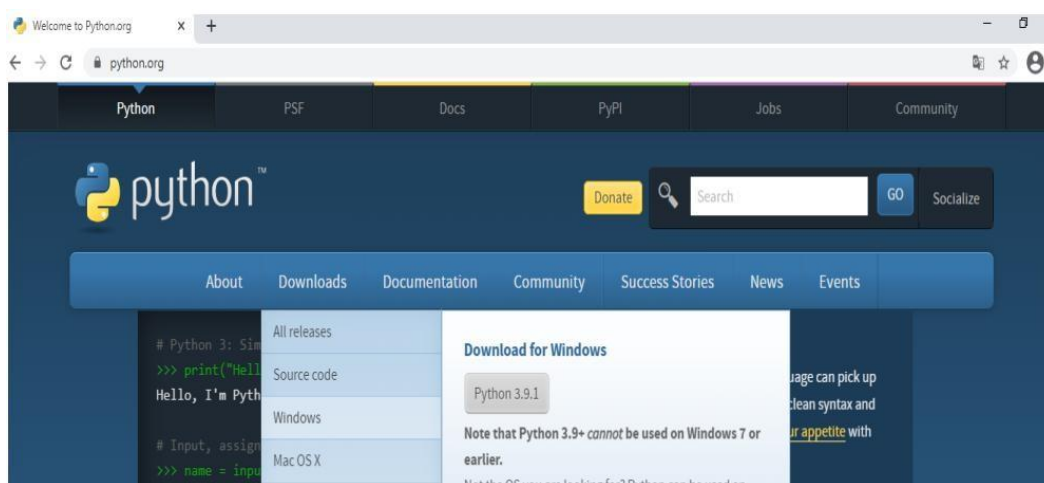
13. Robot texnikada

14. Matematik va ilmiy hisoblashlarni dasturlash

Pythonni katta projeklarda ishlatish mumkin. Chunki, uni chegarasi yo'q, imkoniyati yuqori. Shuningdek, u sodda va universalligi bilan dasturlash tillari orasida eng yaxshisidir.

Python dasturini kompyuterga o'rnatish bir necha bosqichlardan iborat.

<https://www.python.org/downloads/windows/> orqali rasmiy veb saytiga kirib, kompyuteringizning texnik parametrlarini hisobga olgan holda eng so'nggi versiyasini yuklab olamiz. Kompyuterning texnik parametrlariga, razryadi, (64 bit yoki 32 bit) qaysi operatsion sistema o'rnatilganligi va boshqa parametrlar kiradi. Bizhozir sizga 64 bitli kompyuter uchun o'rnatish jarayonini tushuntirib o'tamiz.



1-rasm

Yuklab olib o‘rnatishni boshlaymiz. Pastdagi rasmlarda o‘rnatish jarayoni rasmlarketma – ketligida keltirilgan.



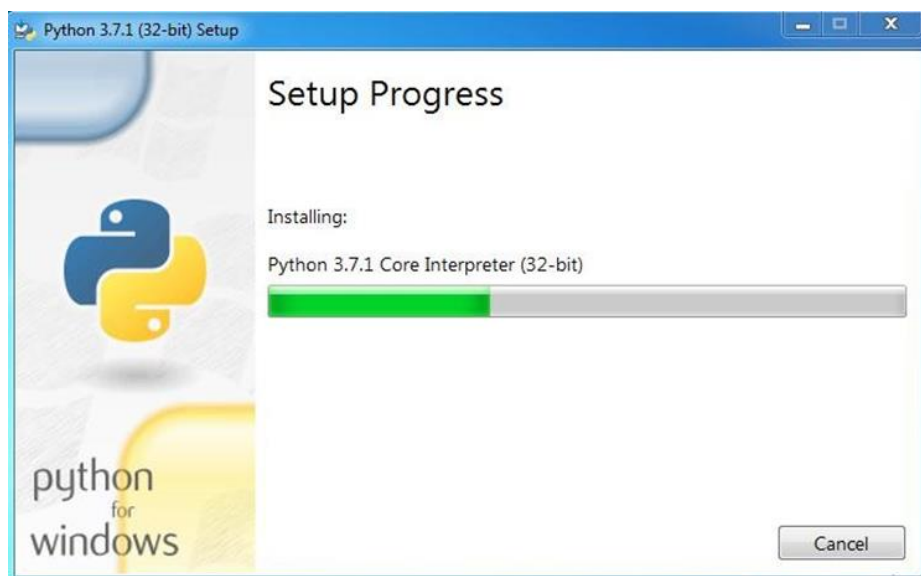
2-rasm

Dastur ustiga sichqonchani ikki marta bosib, o‘rnatishni boshlaymiz. Quyidagi oyna hosil bo‘ladi.



3-rasm

Bu oynadan Add Python 7 to Path ga belgi qo‘yib, Install Now ni tanlaymiz.



4-rasm

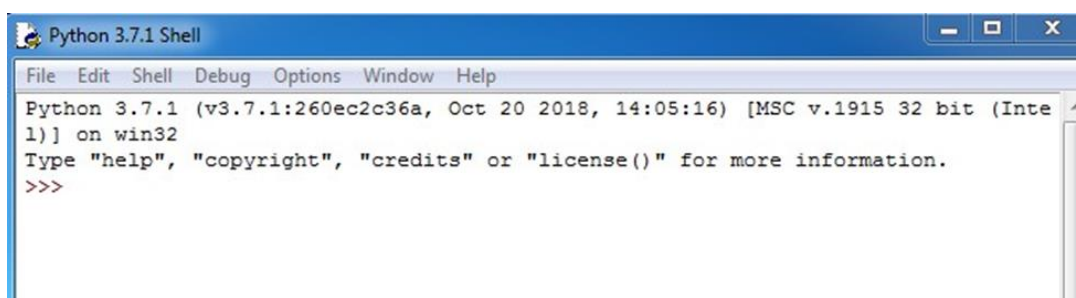
Dastur o‘rnatib bo‘lingach Closeni bosib, ishni yakunlaymiz

Shuni alohida ta'kidlash kerakki, dastur o'rnatib bo'lingach, Python dasturlash tilining ishchi stoliga alohida belgisi hosil bo'lmaydi. Shuning uchun dasturni Pusk orqali ishga tushiramiz. Har doim yangi dastur bilan ishlashda IDLE ni ishga tushiramiz.

IDLE ni ishga tushirish tartibi

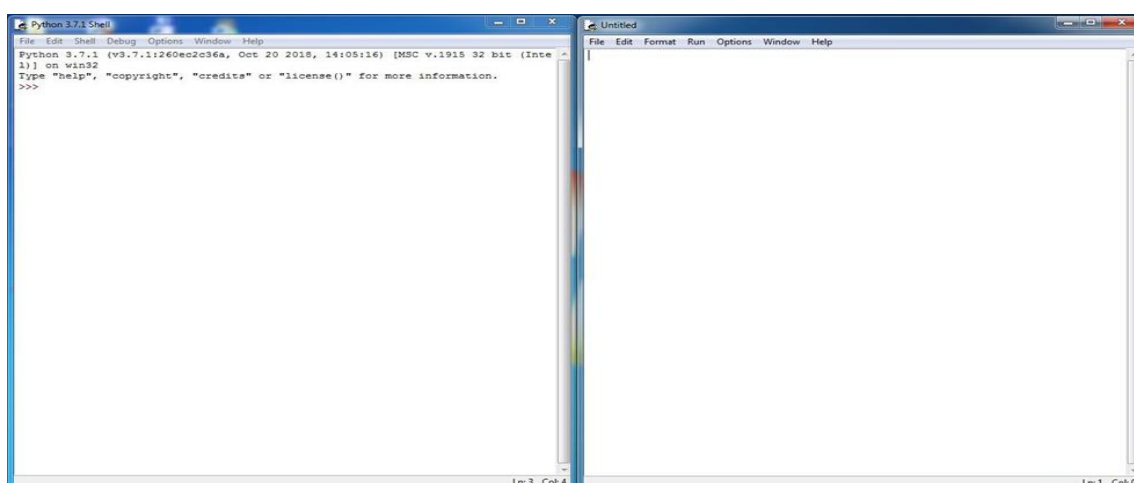
Har doim yangi dastur tuzishda IDLE alohida ishga tushiriladi, ishga tushirish tartibi esa doim bir xil ko'rinishda bo'ladi.

1. Puskdan Python 7 ni topamiz va sichqonchani Python 7 ustiga bosib IDLE ni tanlaymiz.
2. Yangi oq oyna hosil bo'ladi.



5-rasm

File bo'limidan New File ni tanlab (klaviaturadan Ctrl+N), yangi ikkinchi oq oynani hosil qilamiz.



6-rasm

Ikkinchi oyna kod yozish uchun, birinchi oyna esa dastur natijasini ko‘rish uchun ishlatiladi. Unutmang, ikkinchi oynada kodlarni yozib bo‘lgach uni saqlab olishimiz kerak, aks holda dastur ishlamaydi. Saqlash uchun ishchi stolidan “Dasturlar” nomli papka hosil qilib, hamma dasturlarni shu papkaga saqlaymiz. Hozir namuna sifatida biror dastur yozib, uni saqlab ishga tushirishni o‘rganamiz.

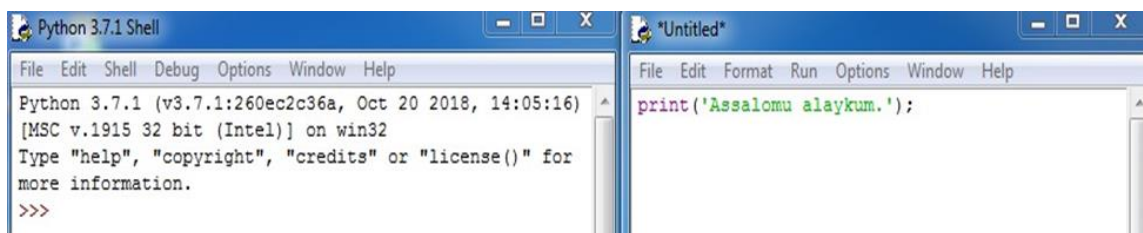
1. Ishchi stolidan “Dasturlar” nomli papka yaratamiz.



7-rasm

1. IDLE ni yuqoridagi tartibda ishga tushiramiz. Ikkinchi oynaga (oxirgi ochilgan oynaga) namunaviy kodlarni yozamiz (bundan keyin kodlarni yozamiz deyilganda ikkinchi oyna nazarda tutiladi.).

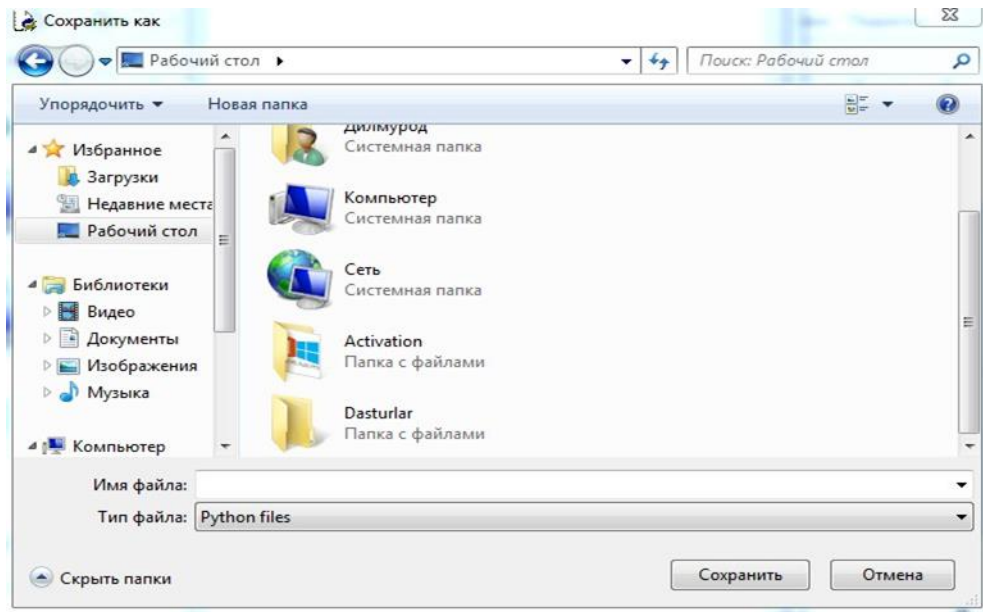
2. Eslatma: dastur kodlarini yozayotganda o‘,g‘ harflaridan foydalanish noqulayliklar tug‘diradi. Shuning uchun bu harflardan foydalanmaymiz.



8-rasm

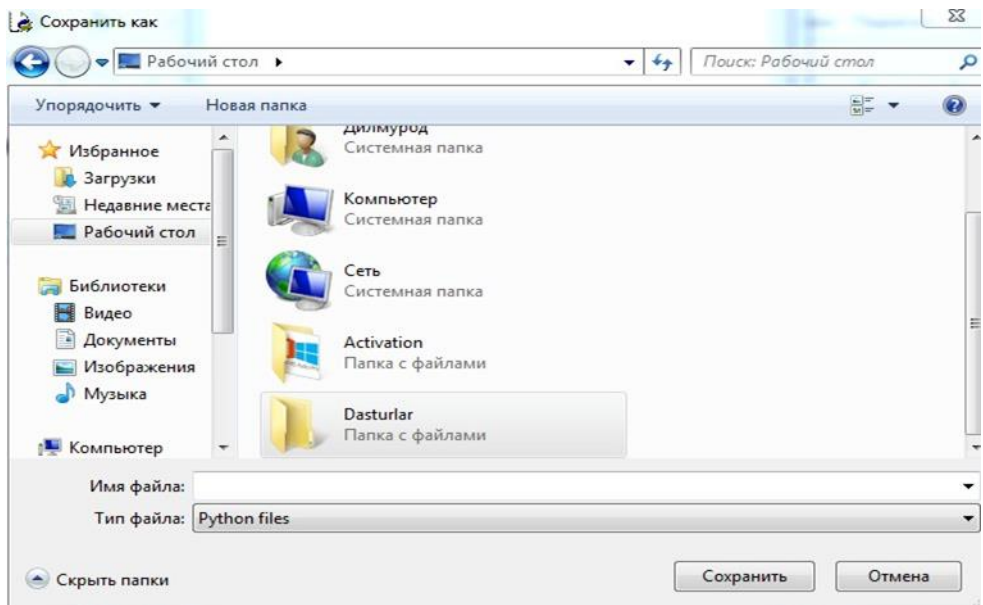
File bo‘limidan Save as ni tanlaymiz.

4. Ochilgan oynadan ishchi stolini tanlaymiz.



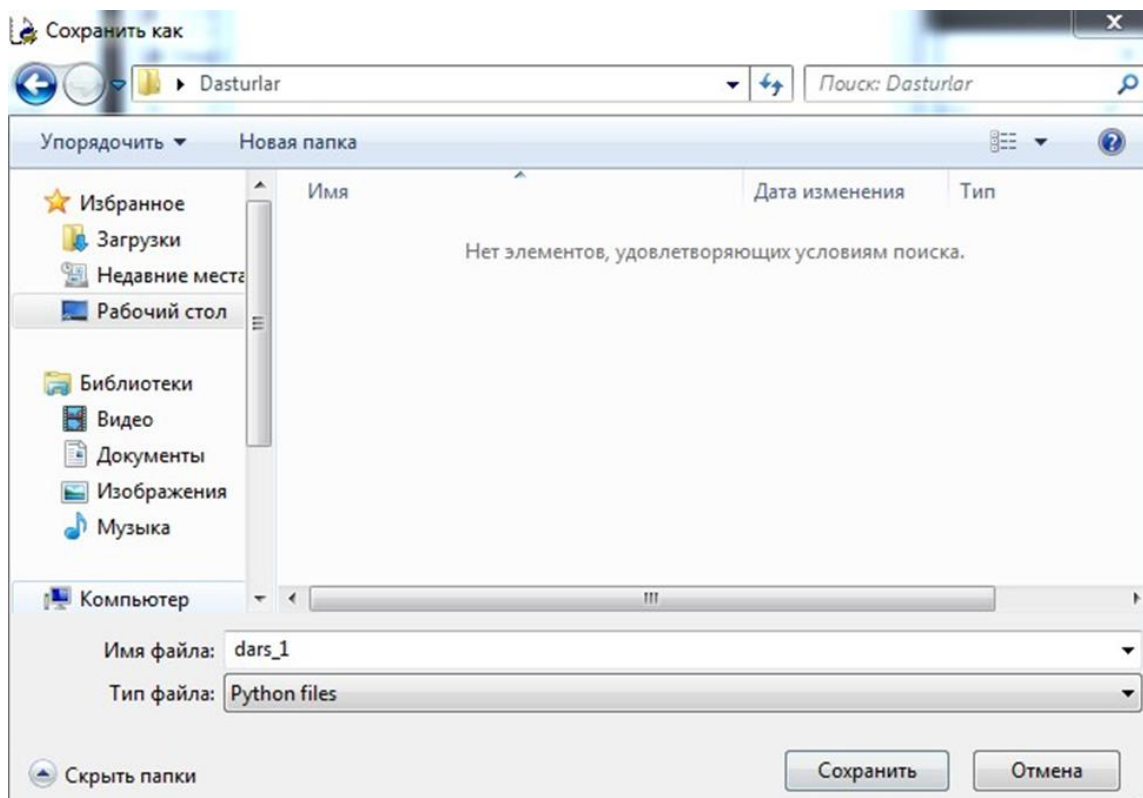
9-rasm

5. Yangi yaratgan Dasturlar nomli papkani tanlaymiz.



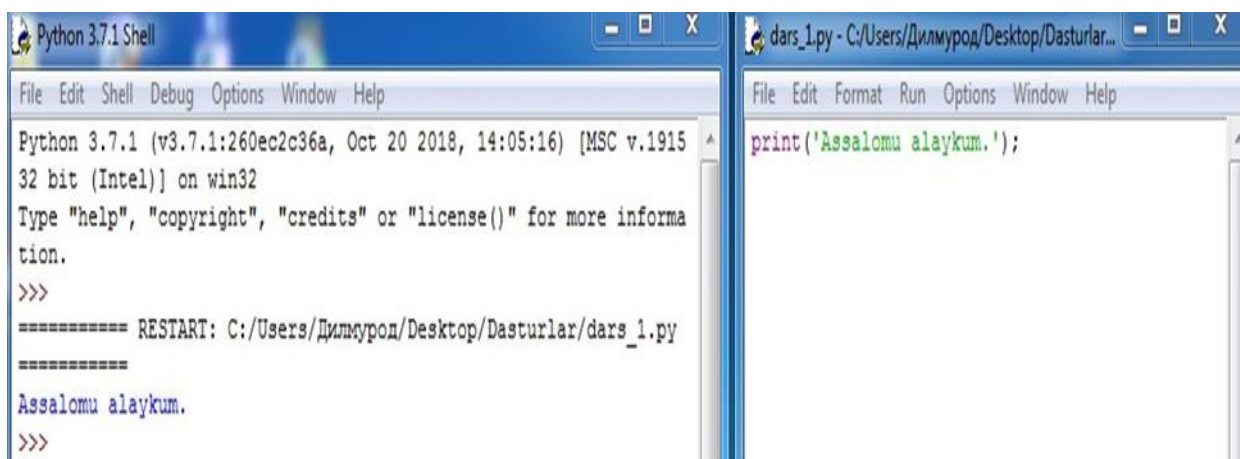
10-rasm

6. Dasturga nom berib saqlaymiz. (masalan dars_1). Nom berishda joy tashlamaslik kerak, uning o'rniga tag chiziq () dan foydalaning



11-rasm

7. Dasturni tekshirish uchun, Run bo‘limidan Run Module ni tanlaymiz. Natija birinchi oq oynada hosil bo‘ladi. (Natija har doim birinchi oynada ko‘rsatiladi.)



12-rasm

Dasturimiz natija berdi, demak kodlarni to‘g‘ri kiritdik. Agar yozilgan dasturga biror o‘zgartirish kiritsak klaviaturadan Ctrl+S klavishlar birikmasini bosgan holda saqlab, yana Run bo‘limi orqali qayta ishga tushirish mumkin.

Python dasturida ishlash tartibi quyidagicha ekan:

1. Puskdan Python 7 ni topib, undan IDLEni ishga tushirish
2. Ochilgan oynaning File bo‘limidan New File ni tanlash (File → New File)
Yangi ochilgan oynaga kodlarni kiritish va saqlash (ishchi stolida yaratilgan yangi papkaga, masalan Dasturlar nomli papkaga) (File → Save as → Ishchi stoli (Рабочи стол) → Yangi yaratilgan papka (Dasturlar) → Dastur nomi (dars_1) → Сохранить (saqlash))
4. Dasturni ishga tushirish (Run → Run Module)
5. Natijani tekshirish 6. Agar dasturga biror o‘zgartirish kiritilsa klaviaturadan CTRL+S klavish birikmalari yordamida qayta saqlash.
7. Har doim yangi dastur yaratilayotganda yuqoridagi tartiblar takroran bajariladi.
8. Keyingi darslarda dasturni ishga tushiramiz deyilganda yuqoridagi tartiblar tushuniladi.
9. Ma’lumotni har safar kiritayotganimizda (dastur natijasini tekshirishda) enter klavishini bosamiz.

PYTHON da kiritish va chiqarish operatorlari

Ma’lumotni konsol ekraniga chiqarish – *print()* funksiyasi hisoblanadi. Funksiyaga argument sifatida konsolga chiqariluvchi qiymatlar (satr, son, ifoda va x.k.) berilishi mumkin:

<pre>print('Hello world!')</pre>	 <pre>Python 3.7.1 Shell File Edit Shell Debug Options Window Help Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14: 1) on win32 Type "help", "copyright", "credits" or "license()" >>> ===== RESTART: D:/Dasturlar/Python/ Hello world! >>></pre>
----------------------------------	--

1-masala. $ax^2+bx+c=0$ kvadrat tenglamaning ildizlarini toping.

Yechish. Kiritiladigan ma'lumotlar – bu tenglama koeffitsienti: a – noma'lumning ikkinchi darajasi oldidagi koeffisient; b – noma'lumning birinchi darajasi oldidagi koeffisient; c – ozod had. Topiladigan natija – x_1 va x_2 tenglama

ildizlari. Buyruqlar: Diskriminantni hisoblash formulasi: $d=b^2-4ac$ Agar diskriminant natijasi noldan katta bo'lsa, u xolda quyidagi formula bilan tenglama ildizlari topiladi:

$$x_1 = \frac{-b - \sqrt{d}}{2a};$$

$$x_2 = \frac{-b + \sqrt{d}}{2a}$$

Agar diskriminant natijasi nolga teng bo'lsa, u xolda quyidagi formula bilan tenglama ildizlari topiladi:

$$x_1 = \frac{-b}{2a}$$

Agar diskriminant natijasi noldan kichik bo'lsa, bu tenglamaning haqiqiy ildizi yo'qligini bildiradi.

```
import math
a=float(input('a='))
b=float(input('b='))
c=float(input('c='))
d=math.pow(b,2)-
4*a*c
if d>0:
    x1=(-b-math.sqrt(d))/(2*a)
    x2=(-b+math.sqrt(d))/(2*a)
    print('x1=',x1,'\nx2=',x2)
elif d==0:
    x=-b/(2*a)
    print('Tenglama bitta ildizga ega\nx=',x)
else: print('Diskriminant 0 dan kichik yechim mavjud emas!')
```

```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 1
1) on win32
Type "help", "copyright", "credits" or "license
>>>
=== RESTART: D:/Dasturlar/Python/Tarmoqlanuvchi
a=1
b=4
c=4
Tenglama bitta ildizga ega
x= -2.0
>>>
=== RESTART: D:/Dasturlar/Python/Tarmoqlanuvchi
a=6
b=4
c=5
Diskriminant 0 dan kichik yechim mavjud emas!
>>>
=== RESTART: D:/Dasturlar/Python/Tarmoqlanuvchi
a=1
b=5
c=4
x1= -4.0
x2= -1.0
>>>
```

MAVZU 2. Sikllar bilan ishlash: sikllar va siklik operatorlar. Sikllar yordamida summa va ko‘paytmalarni topish. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash. Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish..

Reja:

1. Sikllar bilan ishlash: sikllar va siklik operatorlar.
2. Sikllar yordamida summa va ko‘paytmalarni topish.
3. Teylor qatorlari asosida funksiyalarning qiymatini hisoblash.
4. Grafiklar chizish usullari: matplotlib, pyplot bibliotekasi parametrlari, ular yordamida turli shakildagi grafik va gistogrammalarni chizish.

Pythonda siklik operatorlar.

Siklik yoki takrorlanuvchi algoritim yoki dasturning bir necha marta (shu jumladan nol marta) takrorlanishi mumkin bo'lgan qismidir. Har bir siklik tashkil etuvchilar sikl sarlavhasi bilan boshlanadi va yakuniy tasdiq bilan tugaydi. Ularning orasida "sikl tanasi" deb nomlangan operatorlar mavjud. Siklning tanasini tashkil etuvchi buyruqlar (operatorlar) bajarilishining takroriy soni siklni tugatish sharti bilan belgilanadi. To'xtatish sharti sikl parametrlari (sikl o'zgaruvchisi) deb nomlangan maxsus o'zgaruvchining ma'lum bir qiymatiga erishishi yoki ma'lum bir shartning bajarilishi (bajarilishini tugatish) bo'lishi mumkin.

Dasturlash tillarida parametrli sikllarni tashkil qilish uchun FOR ("uchun") operatoridan foydalaniladi va shartli sikllarda WHILE ("aslo") aralash operatoridan foydalaniladi.

Parametrli sikl holatida siklning takrorlanish soni ("takrorlanishlar") oldindan ma'lum va sikl sarlavhasida maxsus ifoda bilan belgilanadi, shartli sikl bo'lsa, har bir keyingi takrorlash bilan siklning tugatish holatini tekshirish kerak.

Agar sikldan tanasida operatorlarni yozish paytida xatolik yuz bersa, siklni tugatish sharti hech qachon bajarilmaydi va sikl cheksiz bo'ladi ("dastur Takrorlanuvchi").

Bir o'lchovli massivlar bilan ishlash uchun parametrli sikllardan foydalanish tavsiya etiladi, chunki takroriy sonlarni sikl boshlanishidan oldin aniqlash mumkin. Bunday holda, massiv elementlarini kiritish uchun parametrli sikl talab qilinadi va sikllar ushbu elementlar bilan har qanday harakatlarni bajarish va natijalarni namoyish qilish uchun ham talab qilinishi mumkin.

Sikllarni biron bir hodisa ro'y bermaguncha ma'lumot kiritiladigan (ba'zi bir manbadan kelgan) vaziyatda ishlatish tavsiya etiladi. Bundan tashqari, barcha ishlov berishlarni ko'pincha massiv yaratmasdan bajarish kerak, chunki elementlarning soni oldindan ma'lum emas.

Oddiy vazifalarni ko'rib chiqing, ularning yechimi siklda hisoblashni talab qiladi.

Odatda sikl operatorlari biror - bir jarayonni qandaydir shart asosida takrorlash uchun ishlatiladi. Python da sikl operatorlarining ikki turi, while va for qaraladi.

while takrorlash operatori

while takrorlash operatori quyidagi umumiy ko'rinishga ega:

while shart ifodasi:

instruksiyalar

while kalit so'zidan keyin shart ifodasi ko'rsatiladi va ushbu shart ifodasi rost

qiymat (True) bo'lar ekan amallar ketma-ketligi takror va takror bajarilishda davom ettiriladi. while operatorining barcha instruksiyalari undan keyingi qarorda

yoziladi va u while kalit so'zidan bitta xat boshi tashlab yoziladi. Masalan:

```
1 sum = 0
2 n = int(input("n="))
3 i = 1
4 while i <= n:
5     sum = sum + i
6     i += 1
7 print("summa(1+...+n) =", sum)
```

Yuqoridagi misolda 1 dan n gacha bo‘lgan sonlar yig‘indisi hisoblash dasturi while operatori yordamida amalga oshirilgan. E`tibor berilsa while operatorining instruktsiyalari undan keyingi qatorda bitta xat boshi tashlab yozilgan. Ushbu holatda while operatori 2 ta instruktsiyalardan tashkil topgan (`sum = sum + i` va `i += 1`).

for takrorlash operatori Yana bir takrorlash operatori – for operatori hisoblanadi. for takrorlash operatori qandaydir sonlar kolleksiyasidagi har bir son uchun chaqiriladi. Sonlar kolleksiyasi `range()` funksiyasi, `list()` funksiyasi yoki [,] qavslarda foydalanuvchi tomonidan shakllantirilgan ro‘yxatlar orqali hosil qilinadi. Quyida for takrorlash operatorining formal aniqlanishi keltirilgan:

```
for int_var in funksiya_range:
```

```
    instruktsiyalar
```

for kalit so‘zidan keyin `int_var` o‘zgaruvchisi (o‘zgaruvchi nomi ixtiyoriy bo‘lishi mumkin) keladi va u faqat butun turdagi qiymatlar qabul qiladi, undan keyin `in` kalit so‘zi (`in` operatori) va `range` funksiyasi chaqirilgan va oxirida “:” belgisi bilan takrorlash operatori asosiy qismi tugaydi. for takrorlash operatorining tana qismi bir yoki bir nechta instruktsiyalardan tashkil topishi mumkin va ular asosiy qismga nisbatan bitta xat boshi tashlab yoziladi.

Takrorlash operatori bajarilganda `range()` funksiyasi hosil qilgan sonlar kolleksiyasidan sonlar ketma-ket `int_var` o‘zgaruvchisiga uzatiladi. Sikl bo‘yicha sonlar ketma-ket o‘tib bo‘lingandan keyin takrorlash operatori o‘z ishiti

tugatadi.

Quyida 1 dan n gacha bo‘lgan sonlar yig‘indisini hisoblash dasturi for operatori yordamida amalga oshirilgan:

```
1 sum = 0
2 n = int(input("n="))
3 for i in range(1,n+1):
4     sum = sum + i
5 print("summa(1+...+n) =", sum)
```

Dastlab konsol ekranidan butun son kiritiladi. Siklda i o'zgaruvchisi aniqlangan bo'lib, u `range()` funksiyasidan qaytarilgan qiymatni o'zida saqlaydi.

Bu erda `range()` funksiyasi 2 ta parametr qabul qilgan. Birinchisi sonlar kolleksiyasini boshlang'ich qiymati va ikkinchisi oxirgi qiymati (oxirgi qiymat

kolleksiyaga kirmaydi). Natijada `range()` funksiyasi $[1, \dots, n-1]$ intervaldagi sonlarni ketma-ket qiymat sifatida qaytaradi va har bir qiymat uchun sikl operatorining tana qismi bajariladi.

`range` funksiyasi. `range` funksiyasining quyidagi shakllari mavjud:

`range(stop)` – 0 dan stop gacha (stop kirmaydi) bo'lgan barcha sonlarni qaytaradi;

`range(start, stop)` – start(kiradi) dan stop (kirmaydi) gacha bo'lgan barcha butun sonlarni qaytaradi;

`range(start, stop, step)` – start (kiradi) dan stop (kirmaydi) gacha bo'lgan barcha butun sonlar step qadam bilan hosil qilinadi va qaytaradi.

Masalan:

1	<code>print(list(range(5)))</code>	<code>#[0, 1, 2, 3, 4]</code>
2	<code>print(list(range(1,5)))</code>	<code>#[1, 2, 3, 4]</code>
3	<code>print(list(range(1,5,2)))</code>	<code>#[1, 3]</code>
4	<code>print(list(range(-5,5,3)))</code>	<code>#[-5, -2, 1, 4]</code>

Bu erda `range(5)` funksiyasi $[0, 1, 2, 3, 4]$ oraliqdagi sonlar kolleksiyasini qaytargan va qaytarilgan sonlarni ro'yxatda jamlash uchun `list()` funksiyasi qo'llanilgan. `range()` funksiyasining boshqa holatlarda qanday qiymatlar hosil qilishini yuqoridagi dasturdan ko'rish mumkin.

Python-da Taylor qatori yordamida funksiyalarni hisoblash va grafik chizish

1. Taylor qatori yordamida funksiyalarni hisoblash

Taylor qatori – funksiyaning qiymatini uning hosilalari orqali yaqinlashib hisoblash usuli. Funksiya $f(x)$ ni nuqta a atrofida Taylor qatori yordamida ifodalash mumkin:

$$f(x) \approx f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots$$

Misol: Sinus funksiyasini Taylor qatori yordamida hisoblash

Sinus funksiyasi uchun Taylor qatori quyidagicha bo‘ladi:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Python kod:

```
import math

def taylor_sin(x, n):
    """
    Sinus funksiyasini Taylor qatori yordamida hisoblash.
    x - burchak radianlarda.
    n - qatorning hadlar soni.
    """
    sin_x = 0
    for i in range(n):
        sign = (-1) ** i # Navbatdagi hadning ishorasi
        term = (x ** (2*i + 1)) / math.factorial(2*i + 1)
        sin_x += sign * term
    return sin_x

# Sin(\pi/4) ni hisoblash
```

```
x_value = math.pi / 4
n_terms = 10 # 10 ta had orqali yaqinlashamiz

approx_sin = taylor_sin(x_value, n_terms)
actual_sin = math.sin(x_value)

print(f"Taylor qatori bo'yicha: {approx_sin}")
print(f"math.sin() natijasi: {actual_sin}")
```

2. Matplotlib yordamida grafik chizish

Matplotlib – Python dasturlash tilida grafiklar chizish uchun eng mashhur kutubxona. Asosan, pyplot moduli orqali ishlatiladi.

Misol: Taylor qatori yordamida sinus funksiyasini yaqinlashtirish va grafik chizish

Python kod:

```
import numpy as np
import matplotlib.pyplot as plt

# Taylor qatori yordamida sinus funksiyasini yaqinlashtirish
x_values = np.linspace(-2*np.pi, 2*np.pi, 400) # -2\pi dan 2\pi gacha oraliq
y_actual = np.sin(x_values) # Asl sinus funksiyasi
```

```
# Turli hadlar soni bilan Taylor qatori hisoblash
```

```
y_approx_3 = [taylor_sin(x, 3) for x in x_values] # 3 ta had
y_approx_5 = [taylor_sin(x, 5) for x in x_values] # 5 ta had
y_approx_10 = [taylor_sin(x, 10) for x in x_values] # 10 ta had
```

Matplotlib yordamida grafik chizish

Matplotlib – Python dasturlash tilida grafiklar chizish uchun eng mashhur kutubxon. Asosan, pyplot moduli orqali ishlatiladi.

Misol: oddiy fizik mayatnik harakatini modellashtirish

Fizikada **kichik burchaklar ostida tebranuvchi mayatnik** ning harakati sinus funksiyasi orqali ifodalanadi. Quyidagi kod mayatnikning tebranishlarini Taylor qatori yordami bilan yaqinlashtiradi va taqqoslaydi.

```
import numpy as np
import matplotlib.pyplot as plt

g = 9.81 # Erkin tushish tezlanishi (m/s^2)
L = 1.0 # Mayatnik uzunligi (m)
t = np.linspace(0, 10, 1000) # Vaqt oralig'i (s)

# Mayatnik burchagini hisoblash (kichik burchaklar ostida yaqinlashish)
def pendulum_theta(t, theta0):
    omega = np.sqrt(g / L) # Burchak chastotasi
    return theta0 * np.cos(omega * t) # Cos orqali yaqinlashtirish

# Turli burchaklar ostida hisoblash
theta_5 = pendulum_theta(t, np.radians(5))
theta_15 = pendulum_theta(t, np.radians(15))
theta_30 = pendulum_theta(t, np.radians(30))

# Grafikni chizish
plt.figure(figsize=(10, 5))
```

```
plt.plot(t, theta_5, label="5° boshlang'ich burchak")
plt.plot(t, theta_15, label="15° boshlang'ich burchak")
plt.plot(t, theta_30, label="30° boshlang'ich burchak")

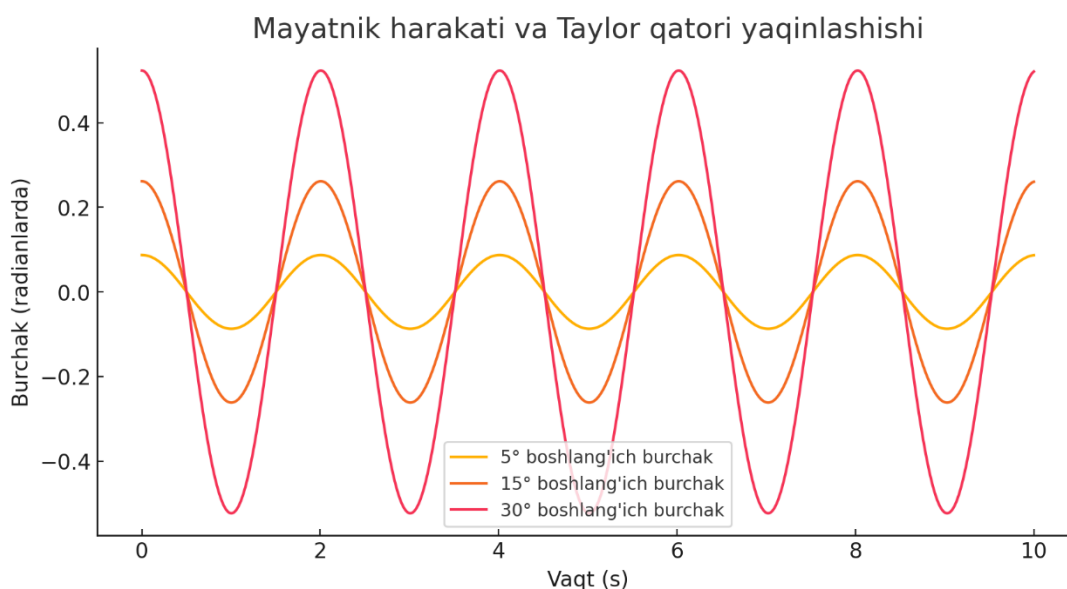
plt.xlabel("Vaqt (s)")
plt.ylabel("Burchak (radianlarda)")
plt.title("Mayatnik harakati va Taylor qatori yaqinlashishi")
plt.legend()
plt.grid()
plt.show()
```

Grafik chizish

```
plt.figure(figsize=(10, 5))
plt.plot(x_values, y_actual, label="Asl sin(x)", linewidth=2)
plt.plot(x_values, y_approx_3, label="3 had", linestyle="dashed")
plt.plot(x_values, y_approx_5, label="5 had", linestyle="dashed")
plt.plot(x_values, y_approx_10, label="10 had", linestyle="dashed")
```

Grafik xususiyatlarini sozlash

```
plt.axhline(0, color="black", linewidth=0.5)
plt.axvline(0, color="black", linewidth=0.5)
plt.legend()
plt.grid()
plt.title("Taylor qatori bilan sin(x) yaqinlashishi")
plt.show()
```



Bu grafik sarka pendulum (oddiy fizik mayatnik) harakatining vaqtga bog‘liq burchak o‘zgarishini ko‘rsatadi. Grafikda turli boshlang‘ich burchaklar (5° , 15° , 30°) ostida mayatnikning tebranishlari tasvirlangan. Agar qo‘shimcha tushuntirish yoki o‘zgarishlar kerak bo‘lsa, xabar bering!

3. Gistogramma chizish

Gistogrammalarni `plt.hist()` yordamida chizish mumkin.

Misol: Tasodifiy sonlar histogrammasi

```
# Tasodifiy normal taqsimlangan sonlar
data = np.random.randn(1000)

plt.figure(figsize=(8, 5))
plt.hist(data, bins=30, color="skyblue", edgecolor="black")
plt.title("Tasodifiy normal taqsimlangan sonlar histogrammasi")
plt.xlabel("Qiymatlar")
plt.ylabel("Chastota")
plt.grid()
plt.show()
```

Bu usullar orqali Taylor qatori asosida funksiyalarni hisoblash va grafiklarni vizualizatsiya qilish bo'yicha amaliy bilimlarga ega bo'lasiz.

Nazorat savollari

1. Sikllardan foydalangan holda 1 dan N gacha bo'lgan sonlarning yig'indisini qanday hisoblash mumkin?
2. Sikl yordamida sonlarning ko'paytmasini hisoblash algoritmini yozing.
3. Bir qator ichida for va while sikllarini ishlatgan holda bir xil natijaga qanday erishish mumkin?
4. Python dasturida ichma-ich sikllar qanday ishlaydi?
5. Sikllar yordamida rekursiv algoritmlarni qanday yozish mumkin?
6. Taylor qatori nima va u qanday ishlaydi?
7. Sinus funksiyasini Taylor qatori yordamida qanday yaqinlashtirish mumkin?
8. Python dasturida `math.factorial()` funksiyasi qanday ishlaydi?
9. `numpy.linspace()` funksiyasining vazifasi nima?
10. `matplotlib.pyplot.plot()` funksiyasining asosiy parametrlari nimalardan iborat?
11. Gistogramma chizishda `plt.hist()` qanday ishlatiladi?
12. Taylor qatori yordamida hisoblangan sinus qiymati `math.sin()` natijasiga qanchalik yaqin bo'ladi?
13. Taylor qatori yondashuvi nechta hadgacha aniq natijalarni beradi?
14. Python dasturida funksiya yaratish qoidalarini tushuntiring.
15. Taylor qatori qaysi matematik sohalarda keng qo'llaniladi?

Amaliy mashg'ulot topshiriqlari

16. Sikl yordamida jismning erkin tushish vaqtiga qarab balandligini hisoblang.
17. Harakat tenglamasi yordamida turg'un harakat qilayotgan jismning yo'lini hisoblash dasturini yozing.
18. Modellashtirilgan harmonik tebranish tenglamasini Python kodida qanday ifodalash mumkin?

19. Issiqlik almashinuvi bo'yicha Newtonning sovish qonuni uchun hisoblash algoritmini yozing.

20. Elektr zanjiridagi kuchlanish va tok orasidagi bog'lanishni grafikda tasvirlash uchun Python kodini yozing.

Mavzu 3. Eksperiment ma'lumotlarining statistik tahlili. Sonli integrallash usullari va algoritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari. Ular yordamida aniq integrallarni hisoblash.

Reja

1. Eksperiment ma'lumotlarining statistik tahlili.
2. Sonli integrallash usullari va algoritmlari: to'g'ri burchaklar, trapetsiyalar, parabolalar (Simpson) usullari va algoritmlari.
3. Ular yordamida aniq integrallarni hisoblash.

Quyida sonli integrallash usullari va algoritmlari haqida tushuntirish, ularning fizikaga bog'lanishi, va har bir usul uchun Python kodlari keltirilgan.

1. Sonli Integrallash usullari va algoritmlari

Matematik analizda integralni aniq yechish imkoni bo'lmaganda, sonli integrallash usullari qo'llaniladi. Bu usullar fizikada turli hodisalarni modellashtirishda juda muhimdir.

a) To'g'ri burchaklar usuli (Riemann summasi)

Bu usulda integral maydoni kichik to'g'ri burchaklarga bo'linadi va ularning yuzalarini yig'ish orqali hisoblanadi.

$$I \approx \sum_{i=0}^{n-1} f(x_i) \cdot \Delta x$$

Formula:

Bu usul elektr tok zichligini hisoblashda qo'llanadi.

b) Trapetsiya usuli

Bu usulda integral maydoni trapetsiyalar yordamida hisoblanadi, ya'ni to'g'ri chiziqli interpolatsiya qilinadi.

Formula:

$$I \approx \frac{\Delta x}{2} \sum_{i=0}^{n-1} (f(x_i) + f(x_{i+1}))$$

Bu usul issiqlik o'tkazuvchanligi va bosim taqsimotini hisoblashda ishlatiladi.

c) Simpson usuli (Parabolalar usuli)

Bu usulda integral parabola segmentlari bilan taqribiy hisoblanadi.

Formula:

$$I \approx \frac{\Delta x}{3} \sum_{i=0,2,4,\dots}^{n-2} (f(x_i) + 4f(x_{i+1}) + f(x_{i+2}))$$

Bu usul kvant

mexanikasi va yorug'likning interferensiyasi uchun ishlatiladi.

2. Python'da Sonli Integrallash va Fizikaga Bog'lash

Keling, bu usullarni fizikaga bog'liq holda Python'da qo'llab ko'ramiz.

Misol: Zaryadning to'planishi

Elektrostatikadagi tok kuchining vaqt bo'yicha integrali zaryadni aniqlaydi:

$$Q = \int I(t) dt$$

$I(t) = e^{-0.5t} \sin(2t)$ funksiya bo'yicha 0 dan 5 sekundgacha zaryadni hisoblaymiz.

Python kodini yozamiz.

```

import numpy as np
import matplotlib.pyplot as plt

def f(t):
    return np.exp(-0.5 * t) * np.sin(2 * t)

# Integrallash parametrlari
a, b = 0, 5 # Integral chegaralari
n = 100 # Bo'lish soni
x = np.linspace(a, b, n+1)
dx = (b - a) / n

# To'g'ri burchaklar usuli
riemann = np.sum(f(x[:-1]) * dx)

# Trapetsiya usuli
trapz = np.sum((f(x[:-1]) + f(x[1:])) * dx / 2)

# Simpson usuli
if n % 2 == 1:
    n += 1 # Simpson usuli uchun juft bo'lishi kerak
x = np.linspace(a, b, n+1)
dx = (b - a) / n
simpson = (dx / 3) * np.sum(f(x[0:-1:2]) + 4*f(x[1::2]) +
f(x[2::2]))

print(f"To'g'ri burchaklar usuli: {riemann:.5f}")
print(f"Trapetsiya usuli: {trapz:.5f}")
print(f"Simpson usuli: {simpson:.5f}")

```

```
# Grafik
T = np.linspace(a, b, 1000)
plt.plot(T, f(T), label="I(t) = e^(-0.5t) * sin(2t)")
plt.fill_between(T, f(T), alpha=0.3)
plt.xlabel("t (s)")
plt.ylabel("I (A)")
plt.legend()
plt.show()
```

Yuqoridagi kod elektr tok kuchining vaqt bo'yicha integrali yordamida zaryadning to'planishini hisoblaydi.

Amaliy mashg'ulot topshiriqlari

1. Sonli integrallash nima? Aniq va sonli integrallash o'ziga xos farqlarni tushuntiring.
2. To'g'ri burchaklar, traptsiya va Simpson usullarining matematik formulalarini yozish.
3. Ush asboblarni qanday foydalanish qulayroq muhokama qiling.
4. Python Dasturi Yordamida Integrallash

1-topshiriq: Berilgan funktsiyani sonli integrallash

Funktsiya:

$$I(t) = e^{-0,3t} \cdot \sin(5t)$$

Vazifa:

- To'g'ri burchaklar, trapatsiya va Simpson usullari yordamida 0 dan 5 sekundgacha integrallash.

- Har bir usul bo'yicha natijalarni baholash.
- Natijalarni grafik ko'rinishda tasvirlash.

Kod namunasi yuqorida berilgan, uni ishga tushirib, mahsulotni tahlil qiling.

2-topshiriq: Yangi fizik misolni integrallash

ishlab chiqarish fizik hodisalarni sonli integrallash usullari bilan yeching:

1. Kinetik energiyani:

$$E_k = \int_0^t m v(t) v'(t) dt$$

Funksiya: $v(t)=3t^2-2t+1$ Massasi: $m=2\text{kgm} = 2 \text{ kgm}=2 \text{ k g}$

Integrallash oraligi: $0 \leq t \leq 40 \leq t \leq 40 \leq t \leq 4$

2. Elektrining o'rtacha hisobni to'g'rilash:

$$I_{o'rtacha} = \frac{1}{T} \int_0^T I(t) dt$$

Tok funksiyasi: $(I(t) = 5\sin(3t)$

Davr: $T=2\pi/3$

Issiqlik jarayoni:

$$Q = \int_0^t kA \frac{dT}{dx} dt$$

Harorat gradienti: $(dT/dx=4e^{-0,3t})$

Issiqlik o'tkazuvchanligi: $(k=0,5)$

Maydon: $A=0,02$

Integrallash oraligi: $0 \leq t \leq 6$

1. Har bir usul bo'yicha olingan natijalarni baholang.
2. Qaysi usul eng aniq natija berdi? Nima uchun?

Ushbu usullar fizikada qanday real hal qilish uchun ko'rsatma?

MAVZU 4 Oddiy differensial tenglamalarni sonli yechish usullari va algoritmalri. Xarakat tenglamasini sonli yechish. Mexanik harakatlarni modellashtirish va ularni visuallashtirish usullari

Reja

1. Oddiy differensial tenglamalarni sonli yechish usullari va algoritmalri.
2. Xarakat tenglamasini sonli yechish.
3. Mexanik harakatlarni modellashtirish va ularni visuallashtirish usullari

Oddiy differensial tenglamalarni (ODT) sonli yechish uchun **Euler usuli**, **Runge-Kutta usuli** kabi algoritmlar ishlatiladi. Differensial tenglamalarni sonli yechishning asosiy usuli quyidagicha:

Euler usuli bilan harakat tenglamasini yechish

```
import numpy as np

import matplotlib.pyplot as plt

def euler_method(f, y0, t):
```

"Euler usuli yordamida differensial tenglama yechimi"

```
y = np.zeros(len(t))  
  
y[0] = y0  
  
dt = t[1] - t[0]  
  
for i in range(1, len(t)):  
  
    y[i] = y[i-1] + dt * f(y[i-1], t[i-1])  
  
return y
```

Misol: erkin tushish harakati

```
def motion_eq(v, t):  
  
    g = 9.81 # Erkin tushish tezlanishi  
  
return -g # F=mg -> dv/dt = -g
```

```
# Parametrlar  
  
t = np.linspace(0, 10, 100) # Vaqt oraligi  
  
y0 = 0 # Boshlang'ich tezlik  
  
# Yechildi  
  
y = euler_method(motion_eq, y0, t)
```

```
# Grafik
```

```
plt.plot(t, y, label="Tezlik (Euler usuli)")
```

```
plt.xlabel("Vaqt (s)")
```

```
plt.ylabel("Tezlik (m/s)")
```

```
plt.title("Erkin tushish tenglamasi sonli yechilishi")
```

```
plt.legend()
```

```
plt.grid()
```

```
plt.show()
```

Amaliy mashg'ulot topshiriqlari

1. Taylor qatori yordamida eksponenta funksiyasini yaqinlashish bilan hisoblash dasturini yozing.
2. Sinus va kosinus funksiyalarining Taylor qatori asosida qanchalik aniq ishlashini grafik orqali solishtiring.
3. Euler usulidan foydalanib, havosiz sharoitdagi erkin tushish harakatini modellashtiring.
4. Havo qarshiligini hisobga olgan holda jismning erkin tushish harakatini grafikda solishtiring.
5. Newtonning sovish qonuni bo'yicha issiqlikning vaqt bo'yicha o'zgarishini modellashtiring.
6. Ikki jismning elastik to'qnashuvini differensial tenglamalar asosida yechib, harakat trayektoriyasini vizualizatsiya qiling.
7. Massaning prujina bilan tebranishini modellashtirish va grafik orqali tasvirlash.

8. Newtonning ikkinchi qonuni asosida berilgan kuch ostidagi jism harakatini hisoblash va grafikda tasvirlash.
9. Ideal gaz tenglamasidan foydalanib, harorat va bosim orasidagi bog‘liqlikni modellashtirish.

Adabiyotlar

1. Schmidt W., Völschow M. // Numerical Python in Astronomy and Astrophysics. A Practical Guide to Astrophysical Problem Solving, Springer, 2021. — 257 p. — ISBN 978-3-030-70346-2
2. T.Axmadjonov, M. Akramov // Fizik jarayonlarni kompyuterda modellashtirish // T.: “Universitet”, 2024
3. M.Akramov, T.Axmadjanov // Fizik jarayonlarni kompyuterda modellashtirish, o‘quv qo‘llanma, T., 2024
4. Нагаева И.А., Кузнецов И.А. // Основы математического моделирования и численные методы, учебное пособие, Лань.:, М. 2024, 204 с.
5. Слабнов В. Д. // Численные методы, учебное пособие, Лань.:, М. 2024, 392 с.
6. Грабовский, Р.И. Курс физики: Учебник / Р.И. Грабовский. – СПб.: Лань, 2012. – 608 с.
7. Фаритов, Т.А. Курс общей физики: Учебник / Т.А. Фаритов. – СПб.: Лань, 2016. – 656 с.

V. KEYSLAR BANKI

1. Mini Keys: Yorug'lik bosimi va kvant mexanikasi

Vaziyat:

Yorug'likning juda kichik massa zarraga tushishi natijasida unga impul's ta'sir qiladi. Agar fotonning yo'nalishi oldindan ma'lum bo'lsa, demak, uning to'lqin uzunligi ham aniq, ya'ni kvant mexanikasi bo'yicha noaniqlik prinsipiga zid keladi.

Savollar:

- Yorug'lik bosimi qanday o'lchanadi?
- Bu jarayonni Python yoki MATLAB'da qanday modellashtirish mumkin?
- Kvant mexanikasi qonunlariga muvofiqligi qanday tekshiriladi?

2. Mini Keys: Kvant tunnel effekti

Vaziyat:

Elektron past energiyali potensial to'siqdan o'tishi kuzatilmoqda. Klassik fizika bo'yicha bu imkonsiz, lekin kvant mexanikasi bo'yicha elektron tunnel effekti natijasida to'siqdan o'tishi mumkin.

Savollar:

- Tunnel effekti qanday ifodalanadi?
- Python yoki MATLAB yordamida qanday modellashtiriladi?
- Agar to'siq balandligi yoki kengligi o'zgarsa, natija qanday o'zgaradi?

3. Mini Keys: Erkin tushish va gravitatsiya modellashtirish

Vaziyat:

Fazoda joylashgan ikkita jismlarning gravitatsion tortishish kuchi hisoblanadi.

Biror jismning erkin tushish tezlanishi aniqlanadi va bu Nyuton qonunlari bilan solishtiriladi.

Savollar:

- Kompyuterli eksperiment orqali tortishish kuchini qanday hisoblash mumkin?
- Nyuton va Eynshteyn gravitatsion nazariyalari qanday farq qiladi?
- Model natijalari qanday vizuallashtirilishi mumkin?

4. Mini Keys: Elektr zaryadlarining o‘zaro ta’siri

Vaziyat:

Ikki elektr zaryad orasidagi kuch Koulon qonuni asosida hisoblanadi. Agar elektr maydon vektorlari grafikda tasvirlansa, ularning qanday ta’sir qilishi aniq ko‘rinadi.

Savollar:

- Elektr maydon qanday modellashtiriladi?
- Zaryadlarning ta’sir vektorlari qanday o‘lchanadi?
- Kompyuter modeli orqali ushbu jarayonni qanday vizuallashtirish mumkin?

5. Mini Keys: Radiatsiya va issiqlik tarqalishi

Vaziyat:

Infraqizil nurlar bilan isitilayotgan jism atrofga radiatsiya tarqatadi. Stefan-Boltsman qonuniga asosan, radiatsiya intensivligi temperaturaning to‘rtinchi darajasiga bog‘liq.

Savollar:

- Ushbu jarayonni Python yoki MATLAB’da qanday modellashtirish mumkin?

- Agar muhit harorati o'zgarsa, radiatsiya qanday o'zgaradi?
- Model natijalarini qanday vizuallashtirish mumkin?

6. Mini Keys: Kapillyar effekt va sirt tarangligi

Vaziyat:

Suyuqlik tor naycha bo'ylab yuqoriga ko'tariladi. Sirt tarangligi va kapillyarlik hodisasi ta'sirida bu jarayon qanday kechishini modellashtirish kerak.

Savollar:

- Sirt tarangligi qanday fizik qonunlarga asoslanadi?
- Ushbu jarayonni Python yordamida modellashtirish mumkinmi?
- Modellashtirilgan jarayon qanday natijalar beradi?

7. Mini Keys: Doppler effekti va harakatlanuvchi manba

Vaziyat:

Yorug'lik yoki tovush to'lqin manbai harakat qilayotgan taqdirda, qabul qiluvchi signal chastotasi o'zgaradi.

Savollar:

- Doppler effekti qanday modellashtiriladi?
- Python yordamida chastota o'zgarishini qanday hisoblash mumkin?
- Agar manbaning tezligi o'zgarib borsa, natija qanday bo'ladi?

8. Mini Keys: Radioaktiv parchalanish va eksponensial so'nish

Vaziyat:

Modda yadrolari vaqt o'tishi bilan parchalanadi va bu eksponensial qonun asosida modellashtirilishi mumkin.

Savollar:

- Parchalanish doimiysi qanday hisoblanadi?
- Python yordamida radioaktiv parchalanishni modellashtirish mumkinmi?
- Agar dastlabki yadrolar soni ko'paytirilsa, parchalanish jarayoni qanday o'zgaradi?

9. Mini Keys: Yorug'lik interferensiyasi va difraksiyasi

Vaziyat:

Ikki tor teshik orqali o'tayotgan yorug'lik interferensiyasi natijasida to'liqin chiziqlari hosil bo'ladi.

Savollar:

- Interferensiya qanday fizik qonunlarga asoslanadi?
- Python yordamida interferensiya naqshini qanday hosil qilish mumkin?
- Agar yorug'lik to'liqin uzunligi o'zgarsa, naqsh qanday o'zgaradi?

10. Mini Keys: Superpozitsiya va kvant o'xshashliklar

Vaziyat:

Ikkita kvant holati superpozitsiya natijasida yangi holatni hosil qiladi. Agar o'lchash amalga oshirilsa, tizim biror holatga o'tib ketadi.

Savollar:

- Kvant superpozitsiya holatini qanday ifodalash mumkin?
- Python yoki MATLAB yordamida ushbu jarayonni modellashtirish usullari qanday?
- Agar o'lchash erta bajarilsa, natijaga qanday ta'sir qiladi?

TEST SAVOLLARI

1. Model nima?

- A)# Tadqiqot uchun muhim bo'lgan obyekt, jarayon yoki tizimning soddalashtirilgan versiyasi
- B) Haqiqatni to'liq aks ettiruvchi narsa
- C) Faqat matematik formulalar majmuasi
- D) Haqiqiy obyektning fizik nusxasi

2. Modellashtirish qanday turlarga bo'linadi?

- A)# Matematik, fizik, kibernetik
- B) Tabiiy, gipotetik, mutlaq
- C) Biologik, kimyoviy, virtual
- D) Sun'iy, tabiiy, murakkab

3. Fizik modellar qanday bo'ladi?

- A)# Dinamik va statik
- B) Biologik va kimyoviy
- C) Oddiy va murakkab
- D) Matematika va fizika

4. Kompyuter modellashtirish nima?

- A)# Jarayonlarni dasturiy ta'minot orqali tahlil qilish
- B) Faqatgina texnik chizmalarga asoslangan modellar yaratish
- C) Grafik dizayn yaratish
- D) Kompyuterda faqat fizik eksperimentlar o'tkazish

5. Kompyuterli eksperiment bosqichlari qanday?

- A)# Formulani yaratish, hisoblash, natija tahlili
- B) Dastur yozish, grafik chizish, hisobot tayyorlash
- C) Aniq ma'lumot kiritish, modellar yaratish, natijalarni hisoblash

D) Kod yozish, natijani chop etish, ma'lumotlarni tekshirish

6. Hisoblash fizikasi nimani o'rganadi?

A)# Fizik jarayonlarni raqamli modellashtirish

B) Biologik jarayonlarni tahlil qilish

C) Kimyoviy reaksiyalarni modellashtirish

D) Faqat optik hodisalarni tadqiq qilish

7. Modellashtirish fizikasi qaysi fan bilan bog'liq?

A)# Fizika va informatika

B) Biologiya va kimyo

C) Astronomiya va geologiya

D) Iqtisodiyot va falsafa

8. Matematik model nima?

A)# Muammoni yechish uchun matematik formulalar va tenglamalar to'plami

B) Faqat grafik va chizmalar

C) Haqiqiy obyektlarning nusxasi

D) Statistik ma'lumotlar

9. Kompyuterda modellashtirish qaysi vositalar yordamida amalga oshiriladi?

A)# Matlab, Python

B) Word, Excel

C) Photoshop, Illustrator

D) AutoCAD, SolidWorks

10. Modellashtirish uchun operatsion tizim qanday bo'lishi kerak?

A)# Linux yoki Windows

B) Faqat Windows

C) Faqat macOS

D) MS-DOS

11. Matlab qanday dasturiy ta'minot?

A)# Texnik hisob-kitob va vizualizatsiya uchun

B) Faqat matn tahrirlash uchun

C) Veb-sahifalar yaratish uchun

D) Grafik dizayn chizish uchun

12. Spyder nima?

A)# Python uchun ishlatiladigan dasturlash muhiti

B) Web-brauzer

C) Antivir juda kuchli dastur

D) Faqat kompyuter viruslarini aniqlovchi dastur

13. Python qanday dasturlash tili?

A)# Obyektga yo'naltirilgan, universal

B) Faqat internet dasturlar yaratish uchun

C) Faqat grafik modellashtirish uchun

D) Faqat mobil ilovalar yaratish uchun

14. Anaconda nima?

A)# Python dasturiy ta'minoti to'plami

B) Operatsion tizim

C) Sun'iy intellekt dasturi

D) Antivirus dasturi

15. Matlab dasturi qanday paketlarga ega?

A)# Simulink, Image Processing, Control System

B) Photoshop, Premiere Pro, Illustrator

C) MS Office, Excel, Word

D) Autocad, Solidworks, Fusion 360

16. Python qanday maqsadlarda qo'llaniladi?

A)# Ma'lumotlar tahlili, modellashtirish, sun'iy intellekt

B) Faqat veb-sayt yaratish

C) Matn muharriri sifatida

D) Antivirus sifatida

17. Kompyuterda modellashtirish qanday afzalliklarga ega?

A)# Tajribalarning xavfsizligi va tezligi

B) Eksperimentlarni butunlay yo'qotish

C) Haqiqiy tajribalar o'tkazish zarurligi

D) Faqat nazariy tadqiqotlar uchun foydalanish

18. Qaysi dasturlar Python muhitida ishlaydi?

A)# Jupyter Notebook, Spyder

B) Excel, Word

C) Photoshop, Illustrator

D) AutoCAD, SolidWorks

19. Matlab qaysi dasturlash tiliga asoslangan?

A)# C/C++

B) Java

C) Python

D) Pascal

20. Kompyuterli eksperiment qanday natija beradi?

A)# Raqamli model asosida fizik jarayonlarni tahlil qilish

B) Faqat grafik vizualizatsiya yaratish

C) Haqiqiy obyektlarni nusxalash

D) Eksperimentlarni o‘rnini bosuvchi javob topish

21. Python qanday kutubxonalar yordamida modellashtirishga yordam beradi?

A)# NumPy, Matplotlib, Pandas

B) Photoshop, Illustrator, Excel

C) Word, PowerPoint, Outlook

D) SolidWorks, AutoCAD, Maya

22. Matlab'dagi Simulink nima?

A)# Grafik model yaratish vositasi

B) Faqat kod yozish muhiti

C) Sun'iy intellekt ishlab chiqish tizimi

D) Grafik dizayn qilish uchun dastur

23. Jupyter Notebook qanday maqsadda ishlatiladi?

A)# Interaktiv dasturlash muhiti

B) Faqat rasm chizish uchun

C) Web dasturlar yaratish uchun

D) Internetda ma'lumot qidirish

24. Python for Data Science qanday maqsadda ishlatiladi?

A)# Ma'lumotlar tahlili va mashinani o'rganish

B) Faqat sayt yaratish

C) Video tahrirlash

D) Matn yozish

25. Modellashtirish jarayonida qanday omillar muhim?

A)# Aniq ma'lumotlar, algoritmlar, tahlil natijalari

B) Shunchaki tasodifiy tanlangan parametrlar

C) Matematik formulalarga asoslanmasdan tajriba o'tkazish

D) Kod yozishning o‘zi yetarli

26. Kod nima?

A)# Dastur tilidagi buyruqlar to‘plami

B) Faqat raqamlar ketma-ketligi

C) Faqat grafik interfeys

D) Haqiqiy hayotdagi obyekt

27. Algoritm nima?

A)# Muammoni yechish uchun aniq qadamlar to‘plami

B) Faqat dastur kodlari

C) Faqat matematik ifoda

D) Ishlash tezligini oshirish

28. Pythonda o‘zgaruvchanlar qanday ishlaydi?

A)# Dastur bajarilishida qiymatlarni saqlash uchun

B) Faqat dastur boshida aniqlanadi

C) Faqat sonlar bilan ishlaydi

D) Faqat satrlar bilan ishlaydi

29. Ma’lumotlar turlari qaysilar?

A)# Integer, Float, String, Boolean

B) Faqat sonlar

C) Faqat matnlar

D) Faqat haqiqat yoki yolg‘on

30. Python sintaksisining asosiy qoidalari qaysilar?

A)# Indentatsiya, o‘zgaruvchilar e’lon qilish, kod bloklari

B) Faqat kodning uzunligi

C) Har doim vergul ishlatish

D) Faqat matematik amallar

31. Projektning qanday saqlash mumkin?

A) Fayl sifatida saqlash, GitHub'da yuklash

B) Faqat qog'ozda yozib olish

C) Faqat brauzerda ochish

D) Internetda saqlash

32. Python dasturining kengaytmasi qanday bo'ladi?

A) .py

B) .txt

C) .exe

D) .java

33. Dastur yozishda qanday xatoliklar bo'lishi mumkin?

A) Sintaksis xatoliklari, mantiqiy xatolar

B) Faqat klaviatura xatolari

C) Faqat raqamli hisob xatolari D) Faqat grafik xatolar

34. Pythonning asosiy operatorlari qaysilar?

A) +, -, *, /, %

B) Faqat + va -

C) Faqat * va /

D) Faqat % va //

35. Python kodini qanday ishga tushirish mumkin?

A) Terminal yoki IDLE orqali

B) Faqat internet orqali

C) Faqat boshqa dasturlar yordamida

D) Faqat jismoniy qurilmalar orqali

36. If shart operatori qanday ishlaydi?

- A) # Shart bajarilganda kodni ishlatadi
- B) Har doim bajariladi
- C) Faqat matnli ma'lumotlarda ishlaydi
- D) Faqat raqamli ma'lumotlarda ishlaydi

37. Else operatori qanday ishlaydi?

- A) # If sharti bajarilmasa, alternativ kodni ishga tushiradi
- B) Har doim bajariladi
- C) Faqat if bilan birgalikda ishlatiladi
- D) Faqat matematik operatsiyalarda ishlaydi

38. Chala tarmoqli algoritm nima?

- A) # Faqat bitta shart bajarilganda ishlaydigan algoritm
- B) To'liq tarmoqli algoritmdan farqi yo'q
- C) Har doim barcha shartlarni bajaradi
- D) Faqat else bloklari mavjud

39. To'liq tarmoqli algoritm nima?

- A) # If va else shartlaridan foydalanuvchi algoritm
- B) Faqat bitta shartni tekshiradigan algoritm
- C) Faqat murakkab dasturlarda ishlatiladi
- D) Shartsiz ishlovchi algoritm

40. Solishtirish operatorlari qanday?

- A) # ==, !=, <, >, <=, >=
- B) &&, ||, !
- C) +, -, *, /
- D) =, :=, <<, >>

VI. MUSTAQIL TA'LIM MAVZULARI

Mustaqil ishni tashkil etishning shakli va mazmuni

Mustaqil malaka oshirish quyidagi shakllarni o'z ichiga oladi: ochiq o'quv mashg'ulotlari va mahorat darslarini tashkil etish; iqtidorli va iste'dodli talabalar bilan ishlash; ilmiy konferensiyalarda ma'ruza bilan qatnashish; ilmiy jurnallarda maqolalar chop etish; ko'rgazma va tanlovlarda ishtirok etish; ilmiy loyihalarda ishtirok etish; xalqaro (impakt-faktorli) nashrlarda maqolalar e'lon qilish; ixtiro (patent), ratsionalizatorlik takliflari, innovatsion ishlanmalarga mualliflik qilish; monografiya, mualliflik ijodiy ishlar katalogini tayyorlash va nashrdan chiqarish; o'quv adabiyotlari (darslik, o'quv qo'llanma, metodik qo'llanma)ni tayyorlash va nashrdan chiqarish; falsafa doktori (PhD) darajasini olish uchun himoya qilingan dissertatsiyaga ilmiy rahbarlik qilish.

Pedagog kadrlarning mustaqil malaka oshirish natijalari elektron portfolio tizimida o'z aksini topadi.

Mustaqil ta'lim mavzulari

1. **Model va modellashtirish** – Modellashtirishning umumiy tushunchasi va ahamiyati.
2. **Fizik modellar va ularning turlari** – Mexanik, elektr va issiqlik modellarni solishtirish.
3. **Kompyuterda modellashtirishning afzalliklari va cheklovlari** – Haqiqiy tajribalar bilan taqqoslash.
4. **Xisoblash fizikasi va uning asosiy tamoyillari** – Fizik jarayonlarni sonli usullar bilan tahlil qilish.
5. **Kompyuterli eksperiment bosqichlari** – Nazariy model yaratishdan natijalarni tahlil qilishgacha bo'lgan jarayon.
6. **Modellashtirish uchun zarur operatsion tizim va dasturiy ta'minot** – Windows, Linux va MacOS tizimlaridagi farqlar.

7. **Spyder va MatlabWork dasturlarining o‘zaro solishtirilishi** – Ularning imkoniyatlari va ishlatilish sohalari.
8. **Python tilining modellashtirish uchun mosligi** – Uning kuchli tomonlari va imkoniyatlari.
9. **Algoritmlar va ularning tuzilishi** – Oddiy algoritmik tuzilmalar va ularning dasturiy ta’minotga tatbiqi.
10. **O‘zgaruvchanlar va ma’lumot turlari** – Python tilida ma’lumot turlarini tahlil qilish.
11. **Chiziqli algoritmlar va dasturlar yaratish** – Oddiy matematik hisob-kitob dasturlarini yaratish.
12. **Tarmoqlanuvchi algoritmlar** – If-else shart operatorlarini amalda qo‘llash.
13. **Munosabat va solishtirish operatorlari** – Katta-kichik va tenglik operatorlarining dasturlashda qo‘llanilishi.
14. **Mantiqiy operatsiyalar va ularning tahlili** – Boolean algebra va dasturlashdagi ahamiyati.
15. **Sikllar va ularning dasturlashda qo‘llanilishi** – While va for operatorlaridan foydalanish.
16. **Sikllar yordamida matematik hisob-kitoblarni bajarish** – Summalarni hisoblash algoritmi.
17. **Taylor qatori yordamida funksiya qiymatini hisoblash** – Matematik model sifatida qo‘llanilishi.
18. **Massivlar va ularning ishlatilishi** – Bir va ikki o‘lchovli massivlarni shakllantirish.
19. **Tasodifiy jarayonlarni modellashtirish** – Tasodifiy sonlarni yaratish va ularni statistik tahlil qilish.
20. **Grafik tasvirlash usullari** – Matplotlib va Pyplot yordamida grafikalar chizish.
21. **Eksperimental ma’lumotlarni statistik tahlil qilish** – O‘rtacha qiymat, dispersiya va regressiya tahlili.

22. **Sonli integrallash usullari** – To‘g‘ri burchaklar, trapetsiya va Simpson usullari.
23. **Oddiy differensial tenglamalarni sonli usulda yechish** – Euler va Runge-Kutta usullari.
24. **Mexanik harakatlarni modellashtirish** – Fizik modellar orqali vizuallashtirish.
25. **Modellashtirish natijalarini vizuallashtirish** – 3D grafikalar va animatsiyalar yaratish.

Ko‘chma mashg‘ulotlarni tashkil etish bo‘yicha ko‘rsatma va tavsiyalar

Ko‘chma mashg‘ulotlar zamonaviy jihozlar hamda innovatsion texnologiyalarni qo‘llab faoliyat yuritayotgan ishlab chiqarish korxonalar va tashkilotlari, oliy ta‘lim muassasalari, iqtisodiyot tarmoqlari, ilmiy-tadqiqot va loyiha-konstruktorlik muassasalarida olib boriladi.

VII. GLOSSARIY

Termin	Tushuntirish	Definition
Model	Tadqiqot yoki amaliyot uchun muayyan ob’ekt yoki jarayonning soddalashtirilgan tasviri.	A simplified representation of a specific object or process for research or practical purposes.
Modellashtirish	Ob’ekt yoki jarayonning modelini yaratish va uni o’rganish orqali tahlil qilish usuli.	The process of creating and analyzing a model of an object or process.
Modellarning klassifikatsiyasi	Modellar statik, dinamik, deterministik, stokastik, fizik, matematik va boshqa turlarga bo’linadi.	Models are classified into static, dynamic, deterministic, stochastic, physical, mathematical, and other types.
Kompyuterli modellashtirish	Kompyuter yordamida modellar yaratish va ularni tahlil qilish usuli.	The method of creating and analyzing models using a computer.
Hisoblash fizikasi	Fizik jarayonlarni kompyuterda hisoblash va tahlil qilish bilan shug’ullanadigan fan sohasi.	A field of science dealing with the computation and analysis of physical processes using a computer.
Modellashtirish fizikasi	Fizik hodisalarni modellashtirish va ularning matematik ifodalarini yaratish.	Modeling physical phenomena and developing their mathematical representations.
Kompyuterli eksperiment	Kompyuter yordamida o’tkaziladigan tajribalar va ularning natijalarini tahlil qilish.	Experiments conducted using a computer and analyzing their results.
Operatsion tizim	Modellashtirish uchun foydalaniladigan Windows, Linux, macOS kabi dasturiy platformalar.	Software platforms such as Windows, Linux, and macOS used for modeling.
Dasturiy ta’minot	Spyder, Matlab, Anaconda, Python for Data Science kabi dasturlar.	Software tools such as Spyder, Matlab, Anaconda, and Python for Data Science.
Python redaktorlari	Python kodini yozish va bajarish uchun ishlatiladigan Spyder,	Environments used for writing and executing Python code,

	PyCharm, VS Code, Jupyter Notebook, Thonny kabi muhitlar.	including Spyder, PyCharm, VS Code, Jupyter Notebook, and Thonny.
Python	Keng qo'llaniladigan yuqori darajali dasturlash tili.	A widely used high-level programming language.
Pip	Python kutubxonalarini o'rnatish va boshqarish uchun ishlatiladigan paket menejeri.	A package manager for installing and managing Python libraries.
Virtual muhit (venv)	Python loyihalarida alohida kutubxona muhitlarini yaratish va boshqarish vositasi.	A tool for creating and managing isolated library environments in Python projects.
List	Python'da ketma-ketlikdagi ma'lumotlarni saqlash uchun ishlatiladigan o'zgaruvchan tuzilma.	A mutable data structure in Python used for storing sequential data.
Tuple	O'zgarmas ketma-ketlikdagi ma'lumotlarni saqlash uchun ishlatiladigan tuzilma.	An immutable sequence data structure in Python.
Dictionary	Kalit-qiyamat juftliklarini saqlash uchun ishlatiladigan ma'lumotlar tuzilmasi.	A data structure that stores key-value pairs.
Set	Takrorlanmaydigan elementlar to'plamini saqlash uchun ishlatiladigan ma'lumotlar tuzilmasi.	A data structure used to store a collection of unique elements.
Funktsiyalar	Kodni qayta ishlatish uchun yoziladigan, aniq bir vazifani bajaradigan bloklar.	Blocks of code written to perform a specific task and enable code reuse.
Lambda-funktsiyalar	Bir qatorli anonim funktsiyalarni yaratish usuli.	A method of creating single-line anonymous functions.
Iteratorlar va Generatorlar	Takroriy jarayonlarni boshqarish va xotira samaradorligini oshirish uchun ishlatiladigan usullar.	Techniques used to manage repetitive processes and improve memory efficiency.

Modullar	Python kodini modul shaklida tashkil qilish va foydalanish imkoniyatini beradigan tuzilma.	A structure that allows organizing and reusing Python code in modular form.
OOP (Obyektga yo'naltirilgan dasturlash)	Python'da klasslar va obyektlar asosida dasturlash paradigmasi.	A programming paradigm in Python based on classes and objects.
Dekoratorlar	Funksiyalarga qo'shimcha xususiyatlar qo'shish uchun ishlatiladigan vosita.	A tool for adding extra functionality to functions.
Exception Handling	Python'da xatolarni (Exceptions) ushlab va ularga ishlov berish usuli.	A method of catching and handling errors in Python.
Chiziqli algoritmlar	Barcha amallar ketma-ket bajariladigan algoritmlar.	Algorithms where all operations are executed sequentially.
Tarmoqlanuvchi algoritmlar	If-else shart operatorlari orqali turli yo'nalishlarda ishlovchi algoritmlar.	Algorithms that operate in different directions using if-else condition operators.
Sikllar	Takroriy jarayonlarni avtomatlashtirish uchun ishlatiladigan while va for operatorlari.	Loops used to automate repetitive processes with while and for operators.
Massivlar	Bir yoki ikki o'lchovli ma'lumotlar tuzilmalari.	One-dimensional or two-dimensional data structures.
Tasodifiy jarayonlar	Matematik model asosida tasodifiy sonlar bilan ishlash.	Working with random numbers based on a mathematical model.
Grafiklar	Matplotlib, Pyplot yordamida grafik va gistogrammalar chizish.	Drawing graphs and histograms using Matplotlib and Pyplot.
Statistik tahlil	Eksperiment ma'lumotlarini statistik metodlar bilan tahlil qilish.	Analyzing experimental data using statistical methods.

Sonli integrallash	To‘g‘ri burchaklar, trapetsiyalar, Simpson usullari bilan aniq integrallarni hisoblash.	Calculating definite integrals using the rectangle, trapezoidal, and Simpson methods.
Differensial tenglamalar	Oddiy differensial tenglamalarni sonli yechish usullari.	Numerical methods for solving ordinary differential equations.
Mexanik harakatni modellashtirish	Jism harakatini sonli yechish va vizuallashtirish usullari.	Numerical solving and visualization methods for modeling mechanical motion.

VIII. ADABIYOTLAR RO‘YXATI

I. O‘zbekiston Respublikasi Prezidentining asarlari

1. Mirziyoev Sh.M. Byuuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. – T.: “O‘zbekiston”, 2017. – 488 b.
2. Mirziyoev Sh.M. Milliy taraqqiyot yo‘limizni qat’iyat bilan davom ettirib, yangi bosqichga ko‘taramiz. 1-jild. – T.: “O‘zbekiston”, 2017. – 592 b.
3. Mirziyoev Sh.M. Xalqimizning roziligi bizning faoliyatimizga berilgan eng oliy bahodir. 2-jild. T.: “O‘zbekiston”, 2018. – 507 b.
4. Mirziyoev Sh.M. Niyati ulug‘ xalqning ishi ham ulug‘, hayoti yorug‘ va kelajagi farovon bo‘ladi. 3-jild.– T.: “O‘zbekiston”, 2019. – 400 b.
5. Mirziyoev Sh.M. Milliy tiklanishdan – milliy yuksalish sari. 4-jild.– T.: “O‘zbekiston”, 2020. – 400 b.

II. Normativ-huquqiy hujjatlar

15. O‘zbekiston Respublikasining Konstitutsiyasi. – T.: O‘zbekiston, 2023.
16. O‘zbekiston Respublikasining 2020-yil 23-sentabrda qabul qilingan “Ta’lim to‘g‘risida”gi Qonuni.
17. O‘zbekiston Respublikasining “Korrupsiyaga qarshi kurashish to‘g‘risida”gi Qonuni.
18. O‘zbekiston Respublikasi Prezidentining 2015-yil 12-iyundagi “Oliy ta’lim muassasalarining rahbar va pedagog kadrlarini qayta tayyorlash va malakasini oshirish tizimini yanada takomillashtirish to‘g‘risida”gi PF-4732-sonli Farmoni.
19. O‘zbekiston Respublikasi Prezidentining 2019-yil 27-maydagi “O‘zbekiston Respublikasida korrupsiyaga qarshi kurashish tizimini yanada takomillashtirish

chora-tadbirlari to‘g‘risida”gi PF-5729-son Farmoni.

20. O‘zbekiston Respublikasi Prezidentining 2019-yil 27-avgustdagi “Oliy ta’lim muassasalari rahbar va pedagog kadrlarining uzluksiz malakasini oshirish tizimini joriy etish to‘g‘risida”gi PF-5789-sonli Farmoni.

21. O‘zbekiston Respublikasi Vazirlar Mahkamasining 2019-yil 23-sentabrdagi “Oliy ta’lim muassasalari rahbar va pedagog kadrlarining malakasini oshirish tizimini yanada takomillashtirish bo‘yicha qo‘shimcha chora-tadbirlar to‘g‘risida”gi 797-sonli Qarori.

22. O‘zbekiston Respublikasi Prezidentining 2019-yil 8-oktabrdagi “O‘zbekiston Respublikasi oliy ta’lim tizimini 2030-yilgacha rivojlantirish konsepsiyasini tasdiqlash to‘g‘risida”gi PF-5847- sonli Farmoni.

23. O‘zbekiston Respublikasi Prezidentining 2020-yil 29-oktabr “Ilm-fanni 2030 yilgacha rivojlantirish konsepsiyasini tasdiqlash to‘g‘risida”gi PF-6097-sonli Farmoni.

24. O‘zbekiston Respublikasi Prezidentining 2021-yil 17-fevraldagi “Sun‘iy intellekt texnologiyalarini jadal joriy etish uchun shart-sharoitlar yaratish chora-tadbirlari to‘g‘risida”gi PQ-4996-son Qarori.

25. O‘zbekiston Respublikasi Prezidentining 2022-yil 28-yanvardagi “2022-2026 yillarga mo‘ljallangan Yangi O‘zbekistonning taraqqiyot strategiyasi to‘g‘risida”gi PF-60-son Farmoni.

26. O‘zbekiston Respublikasi Prezidentining 2023-yil 25-yanvardagi “Respublika ijro etuvchi hokimiyat organlari faoliyatini samarali yo‘lga qo‘yishga doir birinchi navbatdagi tashkiliy chora-tadbirlar to‘g‘risida”gi PF-14-sonli Farmoni.

27. O‘zbekiston Respublikasi Prezidentining 2023-yil 11-sentabrdagi ““O‘zbekiston - 2030” strategiyasi to‘g‘risida”gi PF-158-son Farmoni.

28. O‘zbekiston Respublikasi Prezidentining 2024-yil 21-iyundagi “Aholi va davlat xizmatchilarining korrupsiyaga qarshi kurashish sohasidagi bilimlarini uzluksiz oshirish tizimini joriy qilish chora-tadbirlari to‘g‘risida” PQ-228-son Qarori.

III. Maxsus adabiyotlar

1. Schmidt W., Völschow M. // Numerical Python in Astronomy and Astrophysics. A Practical Guide to Astrophysical Problem Solving, Springer, 2021. — 257 p. — ISBN 978-3-030-70346-2
2. T.Axmadjonov, M. Akramov // Fizik jarayonlarni kompyuterda modellashtirish // T.: “Universitet”, 2024
3. M.Akramov, T.Axmadjanov // Fizik jarayonlarni kompyuterda modellashtirish, o‘ quv qo‘llanma, T., 2024
4. Нагаева И.А., Кузнецов И.А. // Основы математического моделирования и численные методы, учебное пособие, Лань.:, М. 2024, 204 с.
5. Слабнов В. Д. // Численные методы, учебное пособие, Лань.:, М. 2024, 392 с.
6. Грабовский, Р.И. Курс физики: Учебник / Р.И. Грабовский. – СПб.: Лань, 2012. – 608 с.
7. Фаритов, Т.А. Курс общей физики: Учебник / Т.А. Фаритов. – СПб.: Лан П, 2016. – 656 с.

IV. Elektron ta’lim resurslari

8. www.edu.uz.
9. www.aci.uz.
10. www.ictcouncil.gov.uz.
11. www.lib.bimm.uz
12. [www. Ziyonet. Uz](http://www.Ziyonet.Uz)
13. www.sciencedirect.com
14. <http://www.msu.ru/>