

Date: 17th November-2025

DASTURLAR YARATISH VOSITALARI VA ULARNI O'RNATISH

Xoliqova Maxliyo

Buxoro viloyati G'ijduvon tuman 2-son texnikumi Maxsus fan o'qituvchisi

Annotatsiya: Ushbu maqolada zamonaviy dasturlarni yaratish jarayonida qo'llaniladigan vositalar, ularning funksional imkoniyatlari hamda dasturiy ta'minotni o'rnatishning texnik bosqichlari keng qamrovda yoritilgan. Matn muharrirlari, integratsiyalashgan dasturlash muhitlari, kompilyatorlar, interpretatorlar, frameworklar, SDKlar va versiya boshqaruvi tizimlarining dastur ishlab chiqishdagi o'rni ilmiy asosda tahlil qilingan. Shuningdek, dastur o'rnatish jarayonining talablari, paket turlari, konfiguratsiya bosqichlari va keng tarqalgan muammolarni bartaraf etish mexanizmlari batafsil ko'rib chiqilgan. Maqola dastur yaratish hamda uni o'rnatish jarayonlarining o'zaro uzviy bog'liqligi va tizim samaradorligiga ta'sirini ochib beradi.

Kalit so'zlar: dasturlar yaratish, dasturiy ta'minot, IDE, kompilyator, interpretator, framework, SDK, matn muharriri, testlash, versiya boshqaruvi, Git, o'rnatish jarayoni, konfiguratsiya, dastur paketlari, texnik talablar, dasturlash texnologiyalari.

KIRISH

Zamonaviy jamiyatning raqamli taraqqiyoti dasturiy ta'minotning hayotimizdagi o'rnini yanada muhimlashtirmoqda. Bugungi axborot asrida deyarli barcha sohalar — ta'lim, ishlab chiqarish, sog'liqni saqlash, bank tizimi, transport, kommunikatsiyalar, davlat boshqaruvi va xizmat ko'rsatish tizimlari samarali va barqaror ishlashi uchun dasturiy yechimlarga tayanadi. Ushbu jarayon, o'z navbatida, nafaqat tayyor dasturlardan foydalanishni, balki ularga ehtiyojga qarab yangi dasturlar yaratish, ularni yangilab borish va to'g'ri o'rnatishni ham zaruriyatga aylantirdi.

Dastur yaratish jarayoni murakkab bosqichlardan iborat bo'lib, uning har bir qismi maxsus vosita va texnologiyalardan foydalanishni talab qiladi. Dasturlash tillari bilan bir qatorda integratsiyalashgan dasturlash muhitlari (IDE), kompilyator va interpretatorlar, test muhitlari, versiyalarni boshqarish tizimlari, kutubxonalar hamda SDKlar dasturchilar uchun asosiy platforma bo'lib xizmat qiladi. Bu vositalar kod yozish, testlash, xatoliklarni aniqlash, tizimga moslashtirish va loyihani boshqarish jarayonlarini soddalashtirib, ishlab chiqish samaradorligini sezilarli darajada oshiradi.

Hozirgi kunda dunyoda millionlab dasturiy mahsulotlar mavjud bo'lsa-da, ularning samarali ishlashi ko'p jihatdan to'g'ri o'rnatilishiga bog'liq. Operatsion tizimga mos kelmaydigan versiyani o'rnatish, talablarga javob bermaydigan qurilmaga joylash, qo'shimcha kutubxonalar yetishmasligi, xavfsizlik cheklovlari kabi omillar dastur funksionalligiga salbiy ta'sir ko'rsatishi mumkin. Shu sababli dasturiy ta'minotni o'rnatish jarayoni ham ishlab chiqish jarayoni kabi muhim bo'lib, texnik talablarni chuqur anglashni, konfiguratsiya jarayonlarini to'g'ri boshqarishni taqozo qiladi.



Date: 17th November-2025



Dasturlar yaratish vositalari va ularni o'rnatish bo'yicha chuqur bilimga ega bo'lish nafaqat dasturchilar, balki IT xizmatlari bilan shug'ullanuvchi texnik mutaxassislar, tizim administratorlari hamda o'quvchi-talabalar uchun ham zarurdir. Bu bilimlar yordamida ular zamonaviy talablar asosida yangi dasturiy mahsulot yaratishi, mavjud dasturlarni modernizatsiya qilishi yoki ularni samarali sozlab, tizimda barqaror ishlashini ta'minlashi mumkin.

ASOSIY QISM

Dasturlar yaratish vositalarining mohiyati va ahamiyati. Dasturiy ta'minotni ishlab chiqish jarayoni turli texnik, analitik va ijodiy bosqichlarni qamrab oladi. Jarayonning har bir qismida alohida vositalar qo'llaniladi va ular dasturning sifatini, tezkorligini, moslashuvchanligini hamda xavfsizligini belgilaydi. Dastur yaratish vositalari yordamida kodning mantiqiy tuzilishi shakllantiriladi, xatoliklar aniqlanadi, testlar o'tkaziladi hamda yakuniy natija optimallashtiriladi. Bugungi kundagi ko'plab dasturiy mahsulotlarning yuqori sifatga ega bo'lishi aynan ushbu vositalarning rivojlanganligi bilan bog'liq.

Dasturchilar tomonidan qo'llaniladigan vositalar faqat kod yozish bilan cheklanmaydi; ular loyihani boshqarish, hamkorlik qilish, ma'lumotlarni qayta ishlash, turli platformalar uchun moslashuv yaratish kabi funksiyalarni ham bajara oladi. Shu bilan birga, zamonaviy texnologiyalar sun'iy intellekt asosidagi kod tahlili, avtomatik takliflar va xatolarni aniqlash mexanizmlarini ham taklif qilmoqda.

Dasturlar yaratish vositalarining asosiy turlari.

Matn muharrirlari. Matn muharrirlari dasturlashning eng soddalashtirilgan vositalaridan hisoblanadi. Ular bir nechta dasturlash tillari uchun sintaktik ranglash (syntax highlighting), kodni formatlash, tezkor qidiruv kabi funksiyalarni qo'llab-quvvatlaydi. Ushbu muharrirlar yengil bo'lgani uchun tez ishlaydi va resurslarni kam sarflaydi.

Keng qo'llaniladigan matn muharrirlari:

- **Notepad++** — Windows tizimi uchun eng sodda va keng tarqalgan muharrir.
- **Sublime Text** — yuqori tezlik, plugin qo'llab-quvvatlash va ko'p funktsionalligi bilan mashhur.
- **Vim va Nano** — Linux tizimlari uchun terminal asosidagi kod muharrirlari bo'lib, professional foydalanuvchilar orasida keng qo'llaniladi.

Matn muharrirlari kichik loyihalar, jarayonni tezkor boshlash yoki skriptlar yozish uchun juda qulay.

Integratsiyalashgan dasturlash muhitlari (IDE). IDE — bu dasturchilar uchun barcha zarur vositalarni bir joyda jamlaydigan kompleks platforma. U kod yozish, test o'tkazish, kompilyatsiya qilish, xatolarni topish va loyihani boshqarish jarayonlarini integratsiyalashgan shaklda taqdim etadi. IDE'lar ishlab chiqarish hajmi katta bo'lgan loyihalarda juda muhim rol o'ynaydi.

Eng mashhur IDElar:

- ✓ **Visual Studio** — C#, C++, .NET tillari uchun eng qulay muhitlardan biri.
- ✓ **IntelliJ IDEA** — Java, Kotlin va boshqa JVM tillari uchun mo'ljallangan.

Date: 17th November-2025

- ✓ **PyCharm** — Python dasturchilari orasida eng keng tarqalgan IDE.
- ✓ **Android Studio** — Android mobil ilovalarini yaratish uchun rasmiy muhit.
- ✓ **Eclipse** — ochiq kodli, ko‘p tilli dasturlashga moslangan IDE.

IDE'larning afzalliklari: xatolarni tez aniqlash, kod tahlili, loyiha strukturasini optimallashtirish, Git bilan integratsiya, modul boshqaruvi va ko‘plab qo‘shimcha plugin imkoniyatlari.

Kompilyatorlar va interpretatorlar. Dasturlash tillariga qarab, kod ikki turda bajarilishi mumkin: kompilyatsiya orqali yoki interpretatsiya orqali.

Kompilyatorlar (masalan: GCC, Clang, MinGW, javac) dastur kodini mashina tiliga aylantirib, mustaqil bajariladigan fayl yaratadi. Kompilyatsiyalangan dasturlar juda tez ishlaydi.

Interpretatorlar (masalan: Python interpretatori, Ruby interpretatori, Node.js) esa kodni qadam-baqadam bajaradi. Bu esa moslashuvchanlikni oshiradi va testlashni osonlashtiradi.

Har ikkala usulning afzallik va kamchiliklari mavjud, dasturchilar esa loyihaning talablari asosida mos variantni tanlaydi.

Frameworklar va kutubxonalar. Framework — bu dasturchilar uchun tayyor struktura, modul, funksiyalar to‘plami bo‘lib, murakkab loyihalarni qisqa vaqt ichida yaratishga yordam beradi. Masalan:

- ✓ **Django, Flask** — Python web dasturlash uchun.
- ✓ **React, Angular, Vue.js** — JavaScript uchun interfeys yaratish frameworklari.

- ✓ **Spring Framework** — Java dasturlari uchun eng mashhur platforma.

- ✓ **.NET Framework** — Windows ilovalarini yaratish imkonini beradi.

Frameworklar yordamida kod yozish tezlashadi, xavfsizlik oshadi va loyihalarni modulli tarzda tuzish mumkin bo‘ladi.

SDK (Software Development Kit). SDK — bu ma’lum bir platformaga mos dastur ishlab chiqish uchun mo‘ljallangan vositalar to‘plami. U odatda quyidagilarni o‘z ichiga oladi:

- ✓ kompilyatorlar;
- ✓ kutubxonalar;
- ✓ hujjatlar (documentation);
- ✓ testlash vositalari;
- ✓ namunaviy kodlar.

Masalan:

- **Android SDK** — Android ilovalar uchun;
- **Windows SDK** — Windows dasturlari uchun;
- **iOS SDK** — Apple qurilmalari uchun.

SDKlar orqali platforma bilan bevosita ishlovchi professional darajadagi ilovalar yaratish mumkin.



Date: 17th November-2025

Versiyalarni boshqarish tizimlari. Katta va murakkab loyihalarda kodni boshqarish muhim ahamiyat kasb etadi. **Git** bugungi kunda eng mashhur versiya nazorati tizimidir. U kodning barcha o'zgarishlarini qayd qiladi, jamoaviy ishlashni osonlashtiradi, kodni qayta tiklash imkonini beradi.

Git asosidagi xizmatlar:

- **GitHub**
- **GitLab**
- **Bitbucket**

Ular loyihani omborlarda saqlash, muhokama qilish, xatoliklarni kuzatish va CI/CD texnologiyalaridan foydalanish imkonini beradi.

Testlash va xatoliklarni aniqlash vositalari. Sifatli dastur yaratish uchun testlash jarayoni majburiy sanaladi. Testlash orqali dasturdagi mantiqiy, sintaktik, funksional va xavfsizlikka oid xatoliklar aniqlanadi.

Eng mashhur testlash vositalari:

- ✚ **JUnit** — Java uchun;
- ✚ **PyTest** — Python uchun;
- ✚ **Selenium** — web-ilovalarni testlashda;
- ✚ **Appium** — mobil ilovalar uchun.

Testlash jarayonlari avtomatlashtirilsa, dastur ishonchliligi sezilarli darajada oshadi.

Dasturlarni o'rnatish jarayoni

Dastur o'rnatishning umumiy talablari. Dastur o'rnatishdan oldin tizim quyidagi talablarga javob berishi kerak:

- ❖ operatsion tizim versiyasi;
- ❖ protsessor arxitekturasi;
- ❖ RAM va xotira hajmi;
- ❖ qo'shimcha kutubxonalar yoki frameworklar mavjudligi;
- ❖ internetga ulanish talab qilinsa — barqaror tarmoq.

Agar ushbu talablar bajarilmasa, dastur to'liq ishlamasligi yoki konfiguratsiya xatoliklari yuzaga kelishi mumkin.

O'rnatish paketlari turlari

Har bir operatsion tizimning o'ziga xos o'rnatish formati mavjud:

- **Windows:** .exe, .msi
- **Linux:** .deb (Ubuntu), .rpm (Fedora), .bin
- **MacOS:** .dmg
- **Android:** .apk
- **iOS:** .ipa

Har bir paket turi maxsus skriptlar va konfiguratsiya fayllarini o'z ichiga oladi.

O'rnatish bosqichlari

- **O'rnatish** **faylini** **ishga** **tushirish**

— Jarayon avtomatik ravishda o'rnatish oynalarini ochadi.



Date: 17th November-2025



- **Litsenziya shartlariga rozilik bildirish**
- Ko'pgina dasturlar litsenziya shartnomasini tasdiqlashni talab qiladi.
- **Dastur joylashuvini tanlash**
- Foydalanuvchi papkani mustaqil belgilashi mumkin.
- **Qo'shimcha komponentlarni tanlash**
- Plaginar, kengaytmalar yoki qo'shimcha modullar.
- **Konfiguratsiya bosqichi**
- Tizimga moslashtirish, tilni tanlash, server yoki ma'lumotlar bazasiga ulash.
- **O'rnatishni yakunlash va test ishga tushirish**
- Dastur to'g'ri o'rnatilganini tekshirish.
- **O'rnatishda uchraydigan muammolar va ularni bartaraf etish**
- **Xatolik: kutubxona yetishmaydi**
- Kerakli kutubxona yoki frameworkni o'rnatish.
- **Tizim versiyasi past**
- Operatsion tizimni yangilash.
- **Administrator huquqi yetishmaydi**
- "Run as Administrator" buyrug'idan foydalanish.
- **Fayl shikastlangan**
- Dasturiy paketni rasmiy manbadan qayta yuklash.
- **Dastur yaratish va o'rnatishning o'zaro bog'liqligi.** Dastur qanday vositalar orqali yaratilgan bo'lsa, uning o'rnatish jarayoni ham shu vositalarga bog'liq bo'ladi. Misol uchun:
- Python dasturlari uchun **pip, virtual environment** kabi o'rnatish usullari mavjud.
- Java dasturlari **JDK** va **JRE** orqali bajariladi.
- C# dasturlari **.NET Runtime** talab qiladi.
- Demak, dastur yaratish vositalarini bilmasdan dastur o'rnatish jarayonini to'g'ri tashkil etish qiyin.

XULOSA

Dasturiy ta'minotni yaratish va uni o'rnatish jarayoni zamonaviy axborot texnologiyalari rivojining ajralmas qismi hisoblanadi. Dasturlash vositalarining takomillashuvi nafaqat ishlab chiqish jarayonini soddalashtirib, samaradorlikni oshirmoqda, balki turli darajadagi murakkab dasturiy mahsulotlarning qisqa muddatda yaratilishiga ham imkon bermoqda. Matn muharrirlari, integratsiyalashgan dasturlash muhitlari, kompilyatorlar, interpretatorlar, frameworklar, SDKlar va versiya boshqaruvi tizimlaridan oqilona foydalanish dasturchilar mehnatini yengillashtiradi, kod sifati va loyihaaning barqarorligini ta'minlaydi.

Shuningdek, dasturiy ta'minotni o'rnatish jarayoni uning to'g'ri va samarali ishlashi uchun muhim texnik bosqich sanaladi. O'rnatish talablari, paket turlari, konfiguratsiya jarayonlari, tizimga moslashuvi va xavfsizlik mexanizmlarini inobatga olish dasturda kelib chiqishi mumkin bo'lgan xatoliklarni minimallashtiradi. Dastur yaratish va o'rnatish

Date: 17th November-2025

bosqichlarining o‘zaro uzviy bog‘liqligi shuni ko‘rsatadiki, har bir dasturchi yoki texnik mutaxassis ikki yo‘nalishda ham chuqur bilimga ega bo‘lishi lozim.

Umuman olganda, dasturlar yaratish vositalarini to‘g‘ri tanlash, ularni samarali qo‘llash va dasturiy mahsulotlarni talablarga muvofiq o‘rnatish zamonaviy raqamli ekotizimning barqaror ishlashiga xizmat qiladi. Bu sohadagi bilim va ko‘nikmalarga ega bo‘lish esa IT mutaxassislari, talaba yoshlar, texnik xodimlar hamda dasturiy ta‘minot bilan ishlovchi barcha kasbiy guruhlar uchun zarur bo‘lgan dolzarb kompetensiya sifatida maydonga chiqmoqda.

FOYDALANILGAN ADABIYOTLAR:

1. Ahmedov A. Infokommunikatsiya texnologiyalari. O‘quv qo‘llanma. TATU nashriyoti, Toshkent, 2020, 245 b.
2. Mirzayev Sh. Virtualizatsiya texnologiyalari va bulutli hisoblash asoslari. Fan va texnologiya nashriyoti, Toshkent, 2021, 198 b.
3. Raximov K., Abdullaev N. Axborot texnologiyalariga kirish. Iqtisod-moliya nashriyoti, Toshkent, 2019, 312 b.
4. Soliyev M., Yo‘ldoshev B. Kompyuter tizimlari va tarmoqlar. Innovatsiya nashriyoti, Toshkent, 2022, 280 b.
5. William Stallings. Operating Systems: Internals and Design Principles. Pearson, New York, 2018, 820 p.
6. Tom Clark. Designing and Building Virtualization Solutions. McGraw-Hill Education, New York, 2015, 460 p.
7. VMware Inc. VMware vSphere Documentation. VMware Press, 2023.
8. Microsoft. Hyper-V Virtualization Platform Overview. Microsoft Docs, 2023.
9. Иванов В. Виртуализация вычислительных систем: технологии и решения. Издательство “Питер”, Санкт-Петербург, 2017, 304 с.
10. Соколов А., Трофимов Д. Виртуальные машины и облачные сервисы. Издательство “БХВ-Петербург”, Санкт-Петербург, 2020, 288 с.

