

DASTURIY VOSITALAR ARXITEKTURASINING NAZORATI VA DASTURIY MODULNI ISHLAB CHIQISH TARTIBI

Buriyev Bobur Erkinovich

Alfraganus universiteti Raqamli texnologiyalar kafedrası o'qituvchisi

<https://doi.org/10.5281/zenodo.14545631>

Annotatsiya: Ushbu tezisda dasturiy vositalar arxitekturasining nazorati va dasturiy modulni ishlab chiqish tartibi borasida ayrim mulohazalar yuritilgan. Unda dasturiy vositalar arxitekturasining nazorat turlari, funksional spetsifikatsiyasi, dasturiy modulni ishlab chiqarish jarayonlari va ahamiyati to'g'risida tahlillar asosida fikr yuritilgan.

Kalit so'zlar: imitatsiya, spetsifikatsiya, kompilyatsiya, dasturlash, funksional nazorat, elektron qurilma, algoritim.

Abstract. This thesis presents some considerations on the control of software architecture and the procedure for developing a software module. It discusses the types of control of software architecture, functional specification, software module development processes and their significance based on analysis.

Key words: imitation, specification, compilation, programming, functional control, electronic device, algorithm.

Dasturiy vositalar arxitekturasini nazorat etish uchun aralash nazorat va qo'lda bajariladigan imitatsiya ishlatiladi. Dasturiy vositalar arxitekturasining yuqoridan aralash nazorati – bu uning tashqi bayonini ishlab chiquvchilar: sifat spetsifikatsiyasi ishlab chiquvchilari va funksional spetsifikatsiya ishlab chiquvchilari tomonidan qilinadigan nazoratidir. Dasturiy vositalar arxitekturasining pastdan aralash nazorati – bu uning, dasturiy vositalar tarkibiga, ishlab chiqilgan arxitekturaga mos tarzda kiruvchi, dasturiy ichki tizimning potensial ishlab chiquvchilari tomonidan o'tkaziladigan nazoratidir.

Dasturiy vositalar arxitekturasining qo'lda bajariladigan imitatsiyasi funksional spetsifikatsiyaning qo'lda qilinadigan imitatsiyasi kabi amalga oshiriladi, faqat bu nazoratning maqsadi dasturiy ichki tizimlar o'rtasida o'zaro ishlashni tekshirishdir. Qo'lda bajariladigan imitatsiyadagi kabi funksional spetsifikatsiyada ham avval test tayyorlanishi kerak. Shundan keyin ishlab chiquvchilar guruhi bunday har bir test uchun dasturiy vositalar tarkibiga kiruvchi har bir dasturiy ichki tizimning ishini imitirlashi kerak. Bunda har bir ichki tizim ishini qandaydir bir ishlab chiquvchi (arxitektura muallifi emas), bu ichki tizimning boshqa ichki tizimlar bilan (aniqrog'i, ularni imitirlovchi ishlab chiquvchilar bilan) bo'ladigan hamma o'zaro ishlarini yaxshilab bajarib, dasturiy vositalarning ishlab chiqilgan arxitekturasi mos tarzda, imitirlaydi. Shu bilan tekshirilayotgan arxitektura doirasida butunisicha dasturiy vositalarning imitatsion ishlashi ta'minlanadi.

Dasturiy modulni ishlab chiqishda quyidagi tartibga rioya qilish maqsadga muvofiq:

- modul spetsifikatsiyasini o'rganish va tekshirish, dasturlash tilini tanlash;
- berilganlarning algoritmini va tuzilmasini tanlash;
- modulni dasturlash (kodlash);
- modul matnini silliqlash;
- modulni tekshirish;
- modulni kompilyatsiyalash.

Dasturiy modulni ishlab chiqishning birinchi qadami ko'proq dastur tuzilmasini pastdan aralash nazorat qilishni ifodalaydi: ishlab chiquvchi modulning o'ziga xos xususiyatlarini o'rgana turib, bu xususiyatlar unga tushunarli va u shu modulni ishlab chiqish uchun yetarli ekanligiga ishonch hosil qilishi kerak. Bu qadam oxirida dasturlash tili tanlanadi: u butun dasturiy vositalar uchun oldindan aniqlangan bo'lsa ham, ba'zi bir hollarda, mazkur modulni amalga oshirish uchun ko'proq to'g'ri keladigan boshqa til (agar dasturlash tizimi bunga yo'l qo'ysa), masalan, assembler tili, tanlanishi mumkin.

Dasturiy modulni ishlab chiqishning ikkinchi qadamida qo'yilgan masalani yoki unga yaqin bo'lgan masalani yechishning qandaydir bir algoritmlarining mavjud-mavjudmasligini aniqlash zarur. Agar to'g'ri keladigan shunday algoritm topilsa, undan foydalanish maqsadga muvofiq bo'ladi. Modulning o'z funksiyalarini bajarishda, ishlatiladigan ma'lumotlarning to'g'ri keluvchi tuzilmasini tanlash, ishlab chiqilayotgan modulning mantig'ini va sifat ko'rsatkichlarini ma'lum darajada oldindan aniqlaydi, shuning uchun unga juda ham mas'uliyatli qaror deb qarash kerak.

Uchinchi qadamda tanlangan dasturlash tilida modul matnini tuzish amalga oshiriladi. Modul spetsifikatsiyasida ko'rsatilgan funksiyalarni amalga oshirishda hisobga olinishi kerak bo'lgan detallarning ko'pligi juda ham chalkashib ketgan, bir talay xatoliklar va noaniqliklarga ega bo'lgan matnni yaratishga osonlik bilan olib kelishi mumkin. Bunday modulda xatolarni izlash va unga talab qilingan o'zgarishlarni kiritish juda ham ko'p mehnat talab qiladigan masaladir. Shuning uchun modul matnini tuzishda texnologik asoslangan va amalda tekshirilgan dasturlash fanidan foydalanish juda muhimdir. Birinchi marta bu narsaga Deykstr, tuzilmali dasturlashning asosiy tamoyillarini tuzib va ularni asoslab, e'tiborni qaratgan.

Modulni ishlab chiqishning navbatdagi qadami modul matnini, dasturiy vositalar sifat spetsifikatsiyasiga mos tarzda, tugallangan ko'rinishga olib kelish bilan bog'liq. Ishlab chiquvchi modulni dasturlashda, izohlarni oxirigacha ishlab chiqmasdan va dastur uslubiga qo'yilgan ba'zi bir talablarning buzilishiga yo'l qo'yganda, asosiy e'tiborni modul funktsiyasini to'g'ri amalga oshirilishiga qaratadi. Modul matnini silliqlash paytida, u matnda mavjud bo'lgan izohlarni tahrirlashi va unga, talab qilingan sifatni ta'minlash maqsadida, qo'shimcha izohlarni kiritishi mumkin. Shu maqsadda stilistik talablarni bajarish uchun dastur matni tahrirlanadi.

Modulni tekshirish qadami, modulning ichki mantig'ini, uning yaxshilanishi (uning kompyuterda bajarilishidan foydalanuvchi) boshlanguncha, qo'lda tekshirishni ifodalaydi, dasturlashning, dasturiy vositalar ishlab chiqishning har bir bosqichida qabul qilinadigan qarorlarini nazorat qilishning zarurligi haqidagi texnologiyasini muhokama qilish uchun tuzilgan umumiy tamoyilni amalga oshiradi.

Va, nihoyat, modulni ishlab chiqishning oxirgi qadami modulni tekshirish tugaganligini (kompilyator yordamida) va modulni yaxshilash jarayoniga o'tilganligini bildiradi. Agar biz dasturiy ta'minotni joriy etish bosqichi haqida gapiradigan bo'lsak, unda barqarorlik kabi xususiyat foydalanuvchilar tomonidan tanqidiy emas va ularning nuqtai nazari bo'yicha ahamiyatsiz deb qabul qilinadi, lekin dasturiy mahsulotning tijorat ekspluatatsiyasi bosqichi boshlanishi bilanoq, xarajatlar, uning qo'llab-quvvatlashi va rivojlanishini kuzatish mumkin.

Yuqorida aytilganlarning barchasi funksional moslashuvchan mahsulotlarni yaratishga imkon beradigan ishonchli, yuqori sifatli, samarali va kengaytiriladigan arxitekturaga asoslangan yangi, innovatsion, ilg'or axborot mahsulotlarini yaratishga standartlashtirilgan yondashuvning ahamiyati va zarurligini aniq ko'rsatib turibdi.

References:

1. Maxkamovna, M. N. (2023). SANOAT KORXONALARIDA ISHLAB CHIQRISH VA DISPETCHERLIK BOSHQARUV JARAYONLARINI (SCADA) AVTOMATLASHTIRISH. *ARXITEKTURA, MUHANDISLIK VA ZAMONAVIY TEXNOLOGIYALAR JURNALI*, 2(2), 1-3.
2. Erkinovich, B. B. AVTOMATLASHTIRISH TIZIMLARIDA OBYEKT BARQARORLIGINI NAZORAT QILISH VA MODELASHTIRISHGA DOIR AYRIM MULOHAZALAR.
3. Zaripov, Odiljon, and Shahlo Zaripova. "DINAMIK BOSHQARISH OBYEKTLAR HOLATINI BAHOLASH MASALALARINING DASTURIY TA'MINOTI." *Innovatsion texnologiyalar* 50.02 (2023): 73-77.
4. Шилов С.Н. Реализация инфраструктуры распределенной хеш-таблицы в рамках кластерной системы DNS / С.Н. Шилов, С.Д. Кургалин, А.А. Крыловецкий // Вестник ВГУ. Серия: Системный анализ и информационные технологии. - № 2. — Воронеж: изд-во Воронежского гос. ун-та, 2012.-С. 74-79.
5. Erkinovich, B. B. (2024). OBYEKT MUVOZANATINI ROSTLASHDA AVTOMATLASHTIRISH ALGORITMI VA DASTURIY VOSITA YARATISHNING AVTOMATIK BOSHQARISH FUNKSIYALARI. *Eurasian Journal of Mathematical Theory and Computer Sciences*, 4(1), 54-60.
6. Керниган Б. Язык программирования Си / Б. Керниган, Д. Ритчи // - 2-е изд. - М.: Вильямс, 2007. - 304 с.
7. Кнут Д. Искусство программирования. Сортировка и поиск / Д. Кнут. -Т. 3. - 2-е изд. - М.: Вильямс, 2007. - 824 с.