

Proqramlaşdırma Dillərinin Müqayisəli Təhlili və İnkişaf Tendensiyaları

Şəfəqət Mahmudova

AMEA İnformasiya Texnologiyaları İnstitutu, Bakı, Azərbaycan

shafagat_57@mail.ru

Xülasə— Bu işdə proqram mühəndisliyi və onun əhatə etdiyi bilik sahələri haqqında məlumat verilmişdir. Proqramlaşdırmanın inkişaf mərhələləri və bu sahədə olan problemlər araşdırılmışdır. Proqramlaşdırma dilləri haqqında məlumat verilmiş, onların müqayisəli təhlili aparılmış, üstün və çatışmayan cəhətləri aydınlaşdırılmışdır. Proqramlaşdırma dillərinin reytingi haqqında statistik məlumatlar verilmişdir.

Açar sözlər— proqram mühəndisliyi, proqramlaşdırma, proqramlaşdırma dilləri, reyting cədvəli

I. GİRİŞ

Proqram mühəndisliyi dedikdə standartlardan istifadə etməklə həll edilən məsələnin proqram təminatının işlənilməsi, istismarı, müşayiəti və istifadə edilməsi üçün sistemləşdirilmiş, nizamlanmış metodlara malik bir elm sahəsi başa düşülür [1].

İşlənmiş proqram təminatı texniki, iqtisadi və sosial tələblərə cavab verməlidir. Göründüyü kimi, yüksək keyfiyyətli proqram təminatının yaradılması çox zəhmət tələb edən prosesdir. Burada lazım olan layihənin yüksək keyfiyyətli işlənməsi üçün proseslər, alətlər, texnologiyalar və insan resursları bir-birinə qarşılıqlı təsir göstərməlidir.

Proqram mühəndisliyinin müasir texnologiyalarının əsas məqsədlərindən biri müxtəlif problem yönümlü sahələr üçün proqram məhsullarının (PM) yaradılması, onun bütün həyat dövrünün effektivliyini təmin etməkdən ibarətdir.

Proqram mühəndisliyi aşağıdakı bilik sahələrini özündə birləşdirir [2]:

- Kompüterin əsasları;
- Riyaziyyatın və mühəndisliyin əsasları;
- Peşəkar təcrübə (komanda ilə iş, kommunikasiya bacarıqları, etika);
- Modelləşdirmənin (analiz, tələblərlə iş, spesifikasiyalar) əsasları;
- Proqram təminatının layihələndirilməsi (konsepsiyalar və layihənin strategiyaları, insan-maşın interfeysinin layihələndirilməsi, layihəyələrə dəstək vasitələri);
- Proqram təminatının yoxlanılması və attestasiyası (testləşdirmə, istifadəçi interfeysinin qiymətləndirilməsi, problemlərin analizi və s.);
- Proqram təminatının təkamülünün əsasları;
- Proqram təminatının işlənilmə prosesləri;

- Proqram təminatının keyfiyyəti;
- Proqram layihələrinin idarə edilməsi (idarəetmənin konsepsiyaları, planlaşdırma və layihələrin icrasının izlənilməsi, personalın və konfigurasiyanın idarə edilməsi).

Kompüterlər və proqramlaşdırmaya dair müxtəlif standartlar mövcuddur. Kompüterlər və proqramlaşdırmaya dair standartların hazırlanmasına əsas beynəlxalq təşkilatlar IEEE və ACM ciddi yanaşdılar. 2005-ci ildə Kompüter standartları dərc edilmişdir [3]:

- Kompüter mühəndisliyi (*Computer Engineering*);
- İnformatika (*Computer Science*);
- İnformasiya sistemləri (*Information Systems*);
- İnformasiya texnologiyaları (*Information Technology*);
- Proqram mühəndisliyi (*Software Engineering*) və s.

Bu sahələr üzrə standartlar kifayət qədər dəqiqdir və aydın təsvir edilmişdir.

Qarşıya qoyulan məsələlərin proqram təminatının yaradılması üçün müxtəlif proqramlaşdırma dillərindən istifadə edilir. Proqramlaşdırma kompüter üçün proqramın işlənilməsi prosesidir.

Məşhur isveçrəli alim, proqramlaşdırmanın banisi olan Vırt Niklaus (Wirth Niklaus) proqram haqqında belə demişdir: "Proqramlar = alqoritmlər + verilənlərin strukturu" [4].

Proqramlaşdırmada əsas şərt alqoritmlərin qurulmasıdır. Alqoritm - qarşıya qoyulan məsələni həll etmək üçün yerinə yetirilməsi vacib olan əməliyyatlar (mərhələlər) ardıcılığıdır. Ümumiyyətlə, alqoritm-verilmiş məsələnin həlli üçün lazım olan əməliyyatları müəyyən edən və onların hansı ardıcılıqla yerinə yetirilməsini göstərən formal yazılışdır. Alqoritmlərin tərtib olunması prosesi alqoritmləşdirmə adlanır [5].

Proqramlaşdırma dedikdə geniş spektrdə proqram təminatının yaradılması prosesi başa düşülür. Bura analiz və məsələnin qoyuluşu, proqramın planlaşdırılması və layihələndirilməsi, alqoritmlərin və verilənlərin strukturunun işlənilməsi, proqramların mətnlərinin yazılması, yerinə yetirilməsi, testləşdirilməsi (sınaqdan keçirilməsi), sənədləşdirilməsi, sazlanması (konfigurasiyası), işin tamamlanması və müşayiət edilməsi aiddir.

İlk proqramçı, ingilis riyaziyyatçısı Ada Lavleys hesab olunur. İxtisasca riyaziyyatçı olan bu xanım tarixdə ilk

proqram yazması ilə məşhurdur. Çarlz Bebbiç tərəfindən ixtira olunan hesablama maşını üçün ilk proqramı Ada Lavleys tərtib etmişdir. Bu xanım həmçinin "seykl" və "işçi xana" terminlərini də ilk dəfə tətbiq etmişdir. 1975-1980-ci illərdə Ada Lavleysin adıyla bağlı universal proqramlaşdırma dili işlənib hazırlanmışdır. Onun adının əbədləşdirilməsi məqsədilə 1998-ci ildə Ada mükafatı təsis edilmişdir [6].

II. PROQRAMLASHDIRMA DİLLƏRİ HAQQINDA

Proqramlaşdırma dillərinin inkişaf tendensiyasına keçməzdən əvvəl proqramlaşdırma dillərinin hərəkətverici qüvvələri haqqında məlumat vermək lazımdır. Bunlar aşağıdakılardır:

- təkmilləşdirilməsi;
- səmərəliliyə diqqət yetirilməsi;
- məsələnin mürəkkəbliyinin nəzərə alınması;
- proqramın həyat dövrünün uzadılması və s.

Proqramlaşdırma dillərinin inkişaf tendensiyalarını nəzərdən keçirək [7].

1. İlk proqramlaşdırma dilləri

Bu dillər 1940-1950-ci illərdə yaradılmışdır.

Proqram "aşağı səviyyədə" elementar təlimatların xətt ardıcılığından ibarət idi.

Üstünlükləri: yüksək hesablama effektivliyi.

Çatışmazlıqları: Hesablama mühitindən əhəmiyyətli dərəcədə asılılığı.

Nümunələr: maşın kodları, Assembler dili.

2. Prosedurlu (imperativ) proqramlaşdırma dilləri

Bu dillər 1950-1970 cı illərdə yaradılmışdır.

Hər hansı məsələ onun funksional imkanlarından asılı olaraq bir sıra köməkçi məsələlərə bölünür. Bunlar məsələnin alt prosedurları adlanır. İxtiyari prosedurlu proqramın icrası zamanı alt prosedurlar istənilən vaxt çağrıla bilər. Proqramda qlobal və lokal dəyişənlərdən istifadə olunur.

Üstünlükləri: abstraksiyanın daha yüksək səviyyəsi, asılılığın az olması, geniş uyğunluğa malik olması.

Çatışmazlıqları: Böyük əmək sərf edilməsi, proqram kodunun az effektivliyə malik olması.

Nümunələr: Fortran, ALGOL, PL/1, APL, BPL, COBOL, Pascal, C, Basic.

3. Deklarativ proqramlaşdırma dilləri

Bu dillər 1960 cı illərdə yaradılmışdır.

Proqramın yazılışında bu dildən istifadə edildikdə həyata keçirilməsi lazım olan ardıcılıqlar sadə şəkildə təsvir edilir.

Üstünlükləri: Proqramların verifikasiyasının və testləmənin sadəliyi, riyazi formallaşdırmanın ciddiliyi, abstraksiyanın yüksək dərəcədə olması.

Çatışmazlıqları: Reallaşdırmanın çətinliyi, fundamental riyazi biliklərə ehtiyacın olması.

Nümunələr: LISP (Interlisp, Common Lisp, Scheme), SML, Haskell, Prolog.

4. Funksional proqramlaşdırma dilləri

Bu dillər 1960-cı illərdə yaradılmışdır. Proqramlarda müxtəlif funksiyalardan istifadə edilirdi.

Üstünlükləri: Kompüterin yaddaşının tamamilə avtomatik idarə edilməsi, kodun fraqmentlərinin təkrar istifadəsinin sadə olması, parametrik arqumentli funksiyaların geniş spektrə malik olması (parametrik polimorfizm).

Çatışmazlıqları: Proqramın qeyri-xətti struktura malik olması, effektivliyinin nisbətən aşağı olması.

Nümunələr: SML, CaML, Haskell, Miranda, Hope.

5. Məntiqi proqramlaşdırma dilləri

Bu dillər 1970-cı illərdə yaradılmışdır.

Proqram – məntiqi əməllərdən ibarət qaydaların məcmusundan ibarət idi.

Üstünlükləri: Abstraksiyanın səviyyəsinin yüksək olması, məntiqi yönümlü proqramlaşdırmanın rahat olması; ekspert sistemlərinə tətbiqinin rahat olması və s.

Çatışmazlıqları: Məsələlərin məhdudlaşdırılmış imkana və proqramın qeyri-xətti struktura malik olması.

Nümunələr: Prolog, Mercury.

6. Obyekt yönümlü proqramlaşdırma dilləri

Bu dillər 1970 -1980 -ci illərdə yaradılmışdır.

Proqramda obyektlərin təsviri, məcmuları, onların arasında əlaqələr və qarşılıqlı təsirin üsulları izah edilmişdir.

Üstünlükləri: predmet sahəsinə yaxın olması, modelləşdirmə, hadisələr yönümlü olması, abstraksiyanın yüksək səviyyəsi, təsvirlərin təkrar istifadəsi və s.

Çatışmazlıqları: Proqramların verifikasiyası və testləndirilməsinin çətin olması.

Nümunələr: C++, Visual Basic, C #, Eiffel, Oberon.

7. Ssenarili proqramlaşdırma dilləri

Bu dillər 1990-cı illərdə yaradılmışdır.

Proqramda mümkün ssenarilərdən və koddan təkrar istifadə etmək imkanı vardır.

Üstünlükləri: Predmet sahəsinə yaxın olması; abstraksiyanın yüksək dərəcəsi, yüksək mobillik.

Çatışmazlıqları: Proqramların verifikasiyası və testləndirilməsinin çətin olması, çoxsaylı əlavə təsirlərə məruz qalması.

Nümunələr: VBScript, PowerScript, LotusScript, JavaScript.

8. Paralel proqramlaşdırma dilləri

Bu dillər 1980-ci illərdə yaradılmışdır.

Proqramlar eyni zamanda paralel yerinə yetirilə bilər.

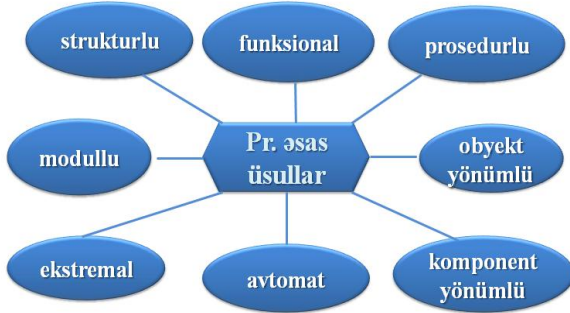
Üstünlükləri: Böyük proqram sistemləri üçün yüksək hesablama effektivliyinə malik olması.

Çatışmazlıqları: Hazırlanan kiçik proqramların maya dəyərini yüksək olması, sürətinin nisbətən aşağı olması və s.

Nümunələr: Ada, Modula-2, Oz

Bunlardan başqa Ekstremal, Aspekt-yönümlü, Avtomat və s. proqramlaşdırma dilləri də vardır.

Proqramlaşdırmada istifadə olunan əsas yanaşmalar şəkil 1 –də göstərilmişdir [8].



Şəkil 1. Proqramlaşdırmada istifadə olunan əsas yanaşmalar

III. MÜASİR PROQRAMLASHDIRMA DİLLƏRİ

Hal-hazırda çoxlu proqramlaşdırma dilləri vardır və bunlardan proqram məhsullarının işlənməsində uğurla istifadə olunur. Buradan belə bir sual meydana çıxır. Yeni proqramlaşdırma dillərinə ehtiyac varmı?

Məsələ ondadır ki, proqramlaşdırma dillərinin güclü və çox funksiyalı olmasına baxmayaraq, hal-hazırda bu dillərin heç birinin sintaksisi mükəmməl və universal deyildir.

Bundan başqa, proqramlaşdırma dilləri daimi inkişaf edir. Çoxnövəli prosessorların yayılması, buludlu proqramlaşdırma, portativ texnikalar (rəqəmli video-kamera, music player, mobil telefonlar, GPS-naviqatorlar, Notebook, İpad və s.), həmçinin paylanmış arxitektura istehsalçılar üçün yeni problemlər yaratmışdır. Artıq mövcud olan proqramlaşdırma dillərinə son zamanlar yeni funksiyaların, paradıqmaların və şablonların əlavə edilməsi onları həddən artıq mürəkkəbləşdirir. Belə hallarda mövcud proqramlaşdırma dillərinə yeni əlavələr etmək əvəzinə sıfırdan başlamaq, yəni, yeni proqramlaşdırma dillərinin işlənməsi daha məqsəduyğundur.

Beləliklə, aşağıda 10 qabaqcıl proqramlaşdırma dili təqdim edilmişdir, belə ki, onların hər birində müəyyən problemləri və ya dillərin spesifik çatışmazlığını həll edə biləcək yeni baxışlar nəzərə alınmışdır. Onlardan bəziləri artıq bitmiş layihədir, bəziləri isə yalnız öz inkişafının ilkin mərhələsindədir.

Qeyd edək ki, onlardan bəziləri populyarlıq qazanmayacaqlar, amma onlardan hər hansı biri proqramlaşdırmada onu əvəz edəcək yeni dillər yaradılmayana qədər inqilabi nailiyyət qazana bilər.

Müasir proqramlaşdırma dillərindən bəziləri aşağıda qeyd edilmişdir:

1. Dart DART

JavaScript interaktivliyin baza elementlərinin əlavə edilməsi və veb sahifələrin yaradılması üçün yaxşıdır, amma proqram kodu minlərlə sətirdən ibarətdir, bu da dilin istifadəsində çətinlik yaradır. Buna görə Google Dartın əvəzedicisi olaraq bu proqramlaşdırma dili yaradıldı —yeni dil veb-proqramlaşdırma dilidir və qeyd olunan çatışmazlıqların aradan qaldırılmasında mühüm rol oynayır.

JavaScript-də, Dart-da C dilində olan oxşar sintaksislər və açar sözlər istifadə olunur. Ancaq bir əhəmiyyətli fərq də var ki, JavaScript prototiplərə əsaslanır, Dart-da isə obyektlər siniflər və interfeyslərin köməyi ilə təyin edilir, Həmçinin Dart proqramçılara C++ və ya Java-da olduğu kimi statik tipli dəyişənləri əlavə etməyə icazə verir [9].

2. Ceylon

Kingin sözlərinə görə, Ceylon, "XML-la Java-nı birləşdirir". Ceylon proqramlaşdırma dili bu sahədə olan problemləri həll etməyə çalışmışdır [10].

3. Go

Go — ümumi təyinatlı proqramlaşdırma dilidir, belə ki, hamı üçün yararlıdır: Adı proqramların işlənməsindən başlayaraq mürəkkəb sistemlərin hazırlanmasına qədər bu dildən istifadə etmək olar. Bu mənada o C və C++-a daha yaxındır. Go müasir funksiyaları özünə daxil edir, lazımsız məlumatlardan yaddaşı təmizləyir və paralelliyi təmin edir [11].

4. F#

1) F# Microsoft şirkəti tərəfindən işlənilmişdir, proqramlaşdırmada funksionallığı və praktikliyi özündə birləşdirir. Çünki F # .Net CLR (Common Language Runtime — ümumdili icra edən mühit)— virtual maşınında birinci sinif dillərə aiddir, o həm də kitabxanaları və funksiyaları dəstəkləyə bilər, başqa CLR dillər kimi [12].

5. Opa

Opa heç bir dili əvəz etmir. O veb-proqramlaşdırmada tamamilə yeni paradıqmanı təşkil edərək onların hamısını birdən əvəz etməyə çalışır. İstifadəçi interfeysi, verilənlər bazası Opa dilində yazılmışdır [13].

6. Fantom

Fantom kross-platforma portativliyi üçün yaradılmışdır. Fantom layihəsi yalnız kompilyatordan ibarət deyil, belə ki necə JVM və ya .Net CLI üçün baytkod-u çıxara bilər, həmçinin API yığımından Java-ı və .Net-i ayırır və portativliyin əlavə səviyyəsini yaradır [14].

7. Zimbu

Öz xarakterinə görə Zimbu proqramlaşdırma dili unikal və spesifikdir, eyni zamanda böyük sayda funksiyaya malikdir. O C-də olan oxşar ifadələr və operatorlardan istifadə edir, amma

onun özünün açar sözləri, verilənlərin tipləri və blok üçün strukturları vardır. O yaddaşı sərbəst şəkildə idarə edir [15].

8. X10

2) X10-da parallel proqramlaşdırma Partitioned global address space (PGAS) (bölünmüş qlobal ünvan fəzasının modelləri) modeli əsasında yaradılmışdır. Bloklarda kod və məlumatlar seçilir və müxtəlif "fəzalar" üzrə bölünür [16].

9. HAXE

Hal-hazırda istifadəçilər haXe-da proqramlar yazı bilər, sonra onları obyekt kodunda JavaScript-də, PHP-də, Flash/ActionScript-də və ya baykod NekoVM-da kompilyasiya edə bilərlər. C# və Java kompilyasiyası üçün əlavə modullar hazırlanmaqdadır [17].

10. Chapel

Chapel Cray Cascade proqramının tərkib hissəsidir, hərbi müdafiə (DARPA) sahəsində məhsuldar hesablamalar üzrə genişmiqyaslı layihə əsasında Amerika İdarəçiliyi tərəfindən qismən maliyyələşdirilmişdir və onlar üçün istifadə olunur. Burada paralel emal alqoritmlərindən istifadə olunur.

Paralel proqramlaşdırma termini kifayət qədər geniş sahəni əhatə edir, belə ki, bir neçə emal etmə qurğusundan ibarət olan hesablama sistemlərində hesablamaların təşkiliylə bağlıdır. Belə sistemlərə çoxnüvəli prosessorlar, ümumi yaddaşlı multiprosessorlu maşınlar, paylanmış yaddaş və ya hibrid arxitekturalara malik məhsuldar hesablama yığınları aiddir.

Son zamanlar paralel hesablamalara böyük diqqət ayrılır. Bu əsasən iki amillə bağlıdır. Birinci amil elmi-texniki proqreslə əlaqədardır, belə ki, riyazi modelləşdirmə metodlarının tətbiqini tələb edən yeni bilik sahələri meydana çıxmışdır. İkinci amil modellərin özlərinin əhəmiyyətli dərəcədə mürəkkəb quruluşla malik olmasıdır.

Şəbəkə texnologiyaları sahəsində əhəmiyyətli irəliləmə ondan ibarətdir ki, onlar paralel hesablamalar üçün müəssisələrin lokal şəbəkələrindən istifadə etməyə imkan verir [18].

IV. POQRAMLAŞDIRMA DİLLƏRİNİN PROBLEMLƏRİ

Müasir dövrdə proqramlaşdırma dillərinin bəzi problemləri aşağıda qeyd edilmişdir.

1. Proqramçıların ən böyük problemlərindən biri paylanmış sistemlərin sayının artması ilə əlaqədar olaraq müxtəlif sistemlər və onların komponentləri arasında əlaqənin təşkilinin çətinləşməsidir. Belə problemləri iki ümumi kateqoriyaya bölürlər: lokal proqramlaşdırma (programming in the small) və qlobal proqramlaşdırma (programming in the big). Lokal proqramlaşdırma sahəsindəki problemlər və onların həlli öz mürəkkəbliyi və təşkilinə görə fərqlənirlər [19].

2. Sistemin dəyişənlərinin tiplərinin problemləri.

3. Meta verilənlərlə əlaqədar problemlər. Metaməlumatlar və ya Metaverilənlər (*Metadata*) — verilənlərin strukturunu və onların emalı metodlarını təsvir edir.

4. Proqramın yerinə yetirilməsi zamanı meydana çıxan problemlər.

5. Qlobal proqramlaşdırmadakı problemlər:

- Adın qoyulması (*naming*) problemi. Əgər coğrafi müxtəlif yerlərdən istehsalçılar onların tərəfindən hazırlanmış siniflərdən təkrarən istifadə etmək istəyirlərsə, eyni adlarla müxtəlif siniflərin yaradılması problemi meydana çıxır.

- Səhvlərin işlənməsi (*error handling*). Səhvlər haqqında informasiyanın təqdim etməsi üçün bir proqramlaşdırma dilində və arxitekturasında qaytarılan qiymətlərdən (*return*) istifadə olunur, xüsusi hallarda (*exception*). Belə dillərin arasında qarşılıqlı təsir üçün müxtəlif dillərin semantik strukturlarının saxlanılmasıyla səhvlər haqqında informasiyanı translyasiya etməyə icazə verən sxemlər və arxitekturalar yaratmaq lazım olur ki, bu da çətinliyə səbəb olur.

- Təhlükəsizlik (*security*). Böyük paylanmış sistemlərin tərtibində əksər hallarda bir neçə proqramlaşdırma dillərindən istifadə olunur ki, bu da müəyyən çətinliklərin əmələ gəlməsinə səbəb olur. Yəni belə hallarda təhlükəsizlik təmin olunmur.

- Versiyaların (*versioning*) nəzarəti. Əksər proqramçılar nə vaxtsa sistem proqram təminatının DLL-kitabxanalarının uyğunsuzluğu problemiylə rastlaşır və onlar anlayırlar ki, arxitekturaların inkişafı əsas problemlərdən biridir.

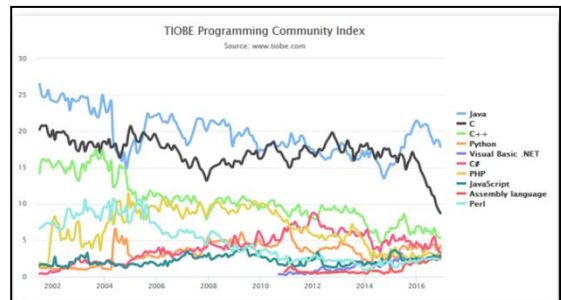
- Miqyaslanma (*scalability*). Bu da əsas problemlərdən biridir. Paylanmış sistemlər təşkilatın daxili şəbəkəsində yüzlərlə istifadəçi üçün əla işləyə bilər, amma Internetdə onlarla milyonlarla istifadəçi işləməyə başladığıqda promer meydana çıxır.

6. Alqoritmlərin həcmnin böyük olması ilə əlaqədar problemlər.

7. Sistemli proqramlaşdırma sahəsində problemlər:

- müasir proqram təminatlarının mürəkkəb olması;
- proqramların verifikasiyası və təhlükəsizliyinin təmin olunması;
- mikroprosessorun mürəkkəb quruluşu;
- çoxnüvəli prosessorlarda problemlərin olması;
- verifikasiyanın paralelləşdirilməsi və s. [18].

Dünyada olan müxtəlif populyar sorgu agentləri tərəfindən proqramlaşdırma dillərinin reyting siyahısı tərtib olunmuşdur [20]. Onlardan ən əsası TIOBE Index mənbəyidir. Şəkil 2 –də TIOBE Index görə proqramlaşdırma dillərinin 2016-cı il üçün reytingini göstərən diaqram verilmişdir :



Şəkil 2. TIOBE Indexə görə proqramlaşdırma dillərinin reytingini göstərən diaqram

V. PROQRAMLAŞDIRMANIN RIYAZİ ASPEKTLƏRİ

Riyaziyyat insanın düşüncəsini inkişaf etdirir. İnkişaf etmiş düşüncə proqramlaşdırmağa kömək edir. Proqramlaşdırmanın riyazi əsaslarının bəziləri şəkil 3 –də göstərilmişdir.



Şəkil 3. Proqramlaşdırmanın riyazi əsasları

Bunların bəziləri haqqında məlumat verək.

Domen adlarının nəzəriyyəsi riyaziyyatın budağı hesab olunur, belə ki, qismən nizanlanmış domenlər adlandırılan çoxluqların xüsusi növlərini öyrənir.

Diskret riyaziyyatda diskret riyazi strukturlar, qraflar və s. öyrənilir. Biz tam ədədlər, qrafiklər və məntiqi ifadələrdən proqramlaşdırmada çox istifadə edirik. Diskret riyaziyyatdan proqram təminatının layihələndirilməsinin spesifikasiyaları üçün istifadə edilə bilər, alqoritmlərin və başqa praktiki tətbiq etmələrin analizi bə s. Bu həqiqətən proqramçı üçün əla vasitədir.

Bunlardan başqa riyazi proqramlaşdırma tətbiqi riyaziyyat elminin əsas bölmələrindən biri olmaqla, onun öyrənmə obyektini mürəkkəb sistem və proseslərin optimal planlaşdırılması və idarə edilməsi kimi məsələlər təşkil edir.

"İntellektual proqramlaşdırma" termini altında elə proqramların işlənilməsi nəzərdə tutulur ki, o proqram insan ağılıyla rahat şəkildə rəqabət apara bilər və ya intellektual məsələlərin həllində ona kömək edə bilər.

İnsan estetik cəhətdən o vaxt təmin olunur ki, o hər şeyi öz əlləriylə yaradır və ustanın əllərinin altından incəsənət əsəri çıxır, bu nöqteyi-nəzərdən bədi tərbiyatına görə proqrama da həminin incəsənət əsəri kimi baxıla bilər [20].

NƏTİCƏ

Qeyd edildiyi kimi proqram mühəndisliyinə aid sahələr çoxdur. Onlardan ən əsası proqramlaşdırma sahəsidir. Proqram təminatı olmadan dünya bazarına çıxmaq mümkün deyil. Müasir proqramlaşdırma dillərindən istifadə etməklə keyfiyyətli proqram təminatı yaratmaq olar. Yuxarıda qeyd edildiyi kimi bu sahədə də çatışmamazlıqlar və problemlər vardır. Bu problemləri həll etməklə rəqabətə davamlı yüksək keyfiyyətli proqram təminatları əldə etmək olar. Bunun üçün bəzi tövsiyələr verilmişdir:

- proqramların yazılışının sadələşdirilməsi;
- proqramların tərtibində hazır komponentlərdən daha çox istifadə edilməsi;

- çoxlu testlərin hazırlanması;
- başqasının proqramı ilə işləmək qabiliyyətinin olması;
- proqramın qısa müddət ərzində icra olunması üçün optimal variantın seçilməsi;
- proqramın həyat dövrünün uzadılması üçün mövcud modellərdən istifadə edilməsi və s.

ƏDƏBİYYAT

- [1] Л. А. Мацяшек, Б. Л. Лионг, Практическая программная инженерия на основе учебного примера, Издательство Бинном: Лаборатория знаний, 2012, 956 с.
- [2] Н. И. Бойко, М.Е. Зверинцев, «Рекомендации по преподаванию программной инженерии и информатики в университетах и др.», Интернет-университет Информационных технологий, 2007, 472 с.
- [3] <https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3>
- [4] https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82_%D0%90%D0%B4%D0%B0
- [5] https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D0%B2%D0%BB%D0%B5%D0%B9%D1%81_%D0%90%D0%B4%D0%B0
- [6] Разные языки программирования и их области применения. Лекция в Яндексе., <https://habrahabr.ru/company/yandex/blog/272759/>
- [7] T. W. Pratt, M. V. Zelkowitz, "Programming languages, design and implementation", Prentice Hall, 2000.
- [8] К. Бакетт, Dart в действии = Dart in Action, ДМК Пресс, 2013, 528 с.
- [9] G. King, Introduction to Ceylon, Part 1, 2011.
- [10] Д. Выкочко, Язык программирования go., Часть 1. Краткий обзор и основы программирования, IBM developer Works, 2011.
- [11] Е. Лазин, М. Моисеев, Д. Сорокин, «Введение в F#», Практика функционального программирования, № 5, 2010.
- [12] 10 языков программирования, которые могут перевернуть мир Т., <https://www.kv.by/content/325498-10-yazykov-programmirovaniya-kotorye-mogut-perevernut-mir-it>
- [13] B. F. Fantom, Dr. Dobb's Journal, 2015.
- [14] Б. Мулинар, «Новый язык программирования», <http://eao197.blogspot.com/2009/10/compprogflame-zimbu.html>
- [15] K. Talha, "OpenFL & Haxe", A Bumpy Start. Gamasutra, 2014, [https://ru.wikipedia.org/wiki/X10_\(%D1%8F%D0%B7%D1%8B%D0](https://ru.wikipedia.org/wiki/X10_(%D1%8F%D0%B7%D1%8B%D0)
- [16] [https://ru.wikipedia.org/wiki/Chapel_\(%D1%8F%D0%B7%D1%8B%D0](https://ru.wikipedia.org/wiki/Chapel_(%D1%8F%D0%B7%D1%8B%D0)
- [17] <http://www.program-code.ru/index.php/net-framework/problemy-programmirovaniya>
- [18] Рейтинг языков программирования в 2016 году., <https://habrahabr.ru/company/kingservers/blog/307012/>
- [19] Tiobe., <http://www.tiobe.com/tiobe-index/>
- [20] Н. Н. Непейвода, И. Н. Григоревский, Е. П. Лилитко, Математические основы программирования, 2014, 16 с., http://psta.psisras.ru/read/psta2014_4_105-121.pdf