

## Obyektyönlü proqramlaşdırmanın əsas konsepsiyaları və xüsusiyyətləri

**Sevinc Cəmil-Cahid qızı Cəbrayılzadə**

*pedaqogika üzrə fəlsəfə doktoru,*

*Azərbaycan Dövlət Pedaqoji Universitetinin dosenti*

**E-mail:** bsevinge.jabrayilzadeh.1973@mail.ru

**Elçin Vahid oğlu İsayadə**

*Azərbaycan Dövlət Pedaqoji Universiteti*

**E-mail:** elchin.isa@bk.ru

**Rəyçilər:** p.ü.f.d., dos. S.S. Həmidov,  
t.ü.f.d. Ç.M. Həməzəyev

**Açar sözlər:** OYP, miras, inkapsulyasiya, polimorfizm, proqramlaşdırma, obyektlər, metodlar, siniflər, sinif nümunələri, xassələr

**Ключевые слова:** ОРР, наследование, инкапсуляция, полиморфизм, программирование, объекты, методы, классы, примеры классов, свойства

**Key words:** OPP, inheritance, encapsulation, polymorphism, programming, objects, methods, classes, class examples, properties

Qeyd edək ki, hazırda müxtəlif mürəkkəblik dərəcələrinə malik sistemlərin işlənilib hazırlanmasında obyektönlü yanaşma qəbul edilir. Prosedur yönümlü dillərin dezavantajı iki problemin olmasıdır: funksiyaların qlobal məlumatlara qeyri-məhdud çıxışı və struktur yanaşmanın əsasını təşkil edən məlumatların və funksiyaların ayrılmasının real dünyanın mənzərəsini zəif əks etdirməsi.

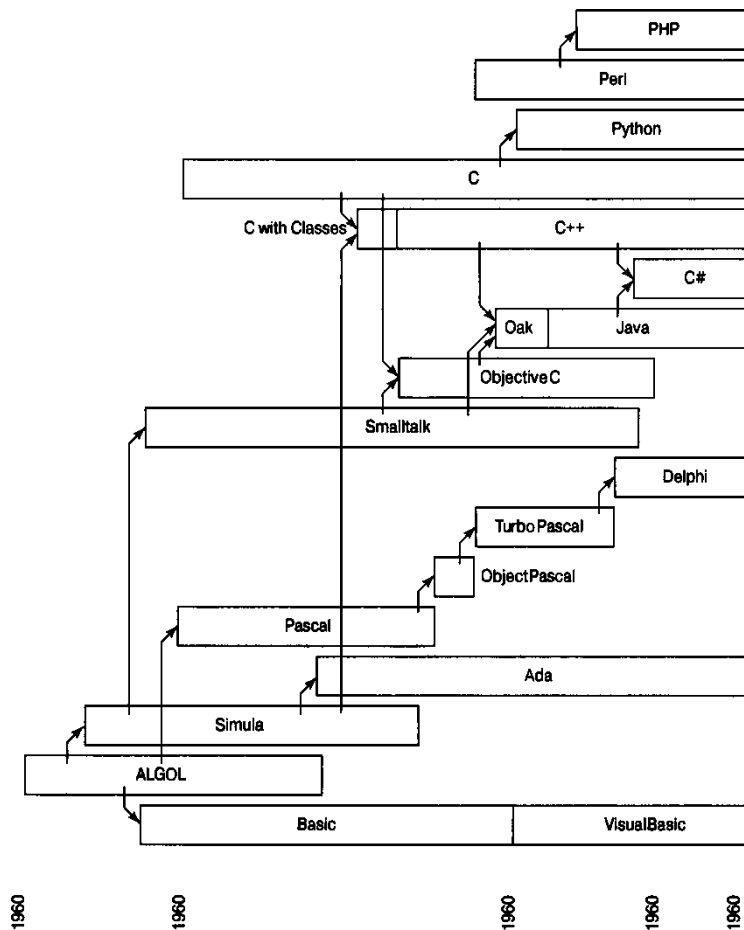
### **OOP-nin yaranması və inkişafı tarixi.**

OOP-struktur proqramlaşdırma üsullarının sənaye proqram məhsullarının artan mürəkkəbliyinin öhdəsindən gəlməyə daha imkan vermədiyi zaman proqram böhranının növbəti mərhələsinə reaksiya kimi ortaya çıxan bir texnologiya kimi başa düşülür. OOP, proqramlaşdırmaya prosessual yanaşmadan və parçalanmaya alqoritmik yanaşmadan istifadə edərək mürəkkəb proqram sistemlərinin yaradılması prosesinə xas olan bəzi problemlərin həlli məqsədi ilə prosedur proqramlaşdırma konsepsiyasının inkişafı kimi yaradılmışdır. Əlbəttə ki, OOP istifadə edərək planlaşdırma prosesi sadələşdirilmişdir, çünki proqramın obyektləri real dünyanın obyektlərinə uyğundur. İlk proqramçılar məşin kodlarından istifadə edərək proqramlar yazdılar. Bu yanaşmanın mürəkkəbliyi səbəbindən insan tərəfindən yazılmış və başa düşülən mətni məşin kodlarına çevirən proqram yaratmaq zərurəti yarandı. Belə proqram “kompilyator”, proqramın mənbə kodunu yazmaq üçün istifadə olunan dil isə “proqramlaşdırma dili” adlanır. İlk tərtibçi “Assembler” idi. Sonradan bir çox müxtəlif proqramlaşdırma dilləri yaradıldı, məsələn: C, ADA, FoxPro, Fortran, Basic, Pascal və s. İnkişafın növbəti mərhələsi obyekt yönümlü proqramlaşdırma idi. Bu paradıqmaya uyğun gələn dillər meydana çıxdı, məsələn: C++, Object Pascal. Proqram təminatının işlənmə sürəti ilə kodun icra sürəti arasında mübarizə aparılmışdır.

Proqramlaşdırmaya prosessual yanaşmadan sonra proqramlaşdırma texnologiyalarının inkişafında növbəti addım OOP-nin meydana çıxması oldu. Proqram kodu artıq “düz” deyil və proqramçı təkə prosedur və funksiyalara deyil, həm də bütün siniflərə tabedir.

Bu gün 2500-dən çox yüksək səviyyəli proqramlaşdırma dili mövcuddur. Bu, xüsusi dillərin müəyyən mövzu sahələrinə yönəldilməsi, eləcə də yeni dillərin meydana gəlməsinin tərtibatçılara getdikcə daha mürəkkəb problemləri həll etməyə imkan verməsi ilə əlaqədardır. 1960-cı illərdə hazırlanmış Simula dili əksər müasir obyekt və obyektlyönlü dillərin ortaq əcdadı hesab olunur. Bu dil ALGOL dilinin ideyalarını inkapsulyasiya və irsiyyət anlayışı ilə tamamlayırdı. Aşağıdakı ən əhəmiyyətli və populyar obyekt və obyektlyönlü proqramlaşdırma dillərinin şəcərəsini göstərir, burada dilin inkişafının intensivliyi düzbucaqlının uzunluğu ilə göstərilir və oxlar bəzi dillərin digərlərinə təsirini göstərir.

Simula dili obyektlər, siniflər, virtual metodlar kimi inqilabi ideyalar təklif edirdi, lakin o zamanlar belə qəbul edilmirdi. Proqramlaşdırmaya yeni, qeyri-prosedural baxış Alan Kay və Dan İnqalls tərəfindən Smalltalk dilində təklif edilmişdir ki, burada “sinif” dil konstruksiyalarının qalan hissəsi üçün əsas ideyaya çevrilmişdir. Məhz bu dil ilk geniş tanınan obyektlyönlü proqramlaşdırma dili hesab olunur. Hal-hazırda OOP paradigmasını həyata keçirən tətbiqi proqramlaşdırma dillərinin sayı digər paradigmaları həyata keçirən dillərin sayından çoxdur. Ən populyar proqramlaşdırma dilləri (C++, Delphi, Java və s.) Simula obyekt modelini həyata keçirir. Smalltalk modelinə əsaslanan dillərə nümunələr Python və Ruby-dir.



Şəkil 1. – Ən mühüm proqramlaşdırma dillərinin şəcərəsi.

**OOP-un əsas anlayışları.**

OOP hər biri müəyyən bir sinfin (xüsusi növün) həyata keçirilməsi olan obyektlər toplusu kimi proqramı təqdim etməyə əsaslanan proqramlaşdırma metodologiyasıdır və siniflər prinsiplərə əsaslanan bir iyerarxiya təşkil edir.

OOP-dan istifadə edərək proqramlar hazırlayarkən, obyektönlü parçalanma istifadə edilə bilər, halbuki əvvəllər proqramlaşdırmaya prosedur yanaşması halında, alqoritmik parçalanma baş verirdi. OOP və strukturlaşdırılmış proqramlaşdırma arasındakı ilk təəccüblü fərq siniflərin istifadəsidir. Sinif məlumat strukturlarını və verilənlərin emal funksiyalarını birləşdirən proqramçı tərəfindən müəyyən edilmiş tiptir. “Sinif” tipli verilənlərin spesifik dəyişənləri sinfin nümunələri və ya obyektləri adlanır.

Sinifdə sahə adlanan sabitlər və dəyişənlər, həmçinin onlar üzərində yerinə yetirilən əməliyyatlar və funksiyalar var. Sinif funksiyaları metodlar adlanır (digər adı üzv funksiyalarıdır). Güman edilir ki, sinif sahələrinə giriş yalnız müvafiq metodları çağırmaqla mümkündür. Sahələr və metodlar sinfin elementləridir (üzvləri). İctimai hissədə yerləşən metodlar sinfin interfeysini təşkil edir və müvafiq sinif obyektini vasitəsilə müştəri tərəfindən sərbəst şəkildə çağırıla bilər. Sinfin özəl bölməsinə yalnız öz metodlarından, qorunan bölməyə isə yalnız öz metodlarından, həmçinin nəsil siniflərin metodlarından daxil olmaq olar. Sinif — “Xassələr, üsullar və hadisələr toplusu”dur. Bu halda “kolleksiya” sinfin fəaliyyətinin metodlar, xassələr və hadisələrlə məcmu şəkildə təmin edilməsi deməkdir.

Obyekt — bir sinfin nümunəsidir. Obyekt bütövlükdə fəaliyyət göstərir, müvafiq sinif hadisələrinə uyğun metodlarla cavab verir.

Hər bir sinifə iki məcburi metodun daxil olması vacibdir: obyekt (konstruktor) yaratmaq, obyektini məhv etmək (dağıdıcı) və obyektin yaradılması prosesi başlatma adlanır. Sinfin yalnız bəzi xüsusiyyətləri ictimai olmalıdır. OOP-da siniflərin rolu ondan ibarətdir ki, onlar həmişə obyektönlü texnologiyaların yaranmasından əvvəl ayrılmış iki funksiyanı yerinə yetirirlər: sinif həm modul, həm də bir növdür. Növ müəyyən dinamik obyektlərin – proqramın icrası zamanı işlənmiş verilənlər elementlərinin statistik təsviridir. OOP sahəsində ən mühüm nailiyyətlərdən biri bəzən dizayn nümunələri adlanan dizayn nümunələrinin metodologiyasıdır.

**OOP-un əsas xassələri.**

OOP-un əsas prinsipləri: qablaşdırma və ya inkapsulyasiya; miras; polimorfizm; mesaj ötürülməsi.

OOP-da irsiyyət onun əsasını təşkil edir. Başa düşmək lazımdır ki, nəsil – başqa siniflərin törədiyi sinif, əcdadın xüsusiyyətlərini miras alır – başqa bir sinifdən yaranan və ya yaranan bir sinif. Ona görə də uşaq həmişə hansı xassələrə malik olduğunu “bilir”, əcdad isə öz övladının xassələrini “bilə” bilməz, çünki o, yeni sinifə əlavə olunacaq xassələri “bilir”. Bu irsiyyət üsulu obyektönlü dillərdə qəbul edilir.

Deməli, mövcud kodu təkrar istifadə etməyə imkan verən irsiyyətdir. Mövcud koddan istifadə etmək üçün proqramçı mövcud siniflər əsasında yeni sinif yaradır. Nəticədə vərəsəlik iki funksiyanı yerinə yetirir: kodun təkrarlanması qarşısının alınması, işin düzgün istiqamətdə inkişafı. Ana sinif və onun nəsiləri arasındakı əlaqə “Vərəsəlik İyerarxiyası” adlanır.

Polimorfizm mənaca oxşar məsələləri müxtəlif yollarla həll etmək üçün əlaqəli obyektlərin xassəsinə aiddir. Məsələn, hərəkət “Qaçmaq” əksər heyvanlar üçün ümumdür. Ancaq onların hər biri fərqli şəkildə işləyir.

Polimorfizm bir hərəkətə vahid ad verilməsidir, daha sonra iyerarxiyadakı hər bir obyekt bu hərəkəti özünə uyğun şəkildə yerinə yetirməklə, obyektlərin iyerarxiyasında yuxarı və aşağı paylaşılır. Qeyd edək ki, “polimorfizm” bir neçə formaya malik olmaq qabiliyyətidir. Ob-

yektyönlü inkişafda bu, icra zamanı müxtəlif tipli obyektlərə əlavə oluna bilən obyektlərə (məlumat strukturlarının elementlərinə) aiddir. Polimorfizm müxtəlif problemlərin həlli üçün eyni funksiyalardan istifadə etməyə imkan verir və bir adın məzmunu obyektin növündən asılı olan müxtəlif hərəkətləri ehtiva etməsində əks olunur. Eyni baza sinfinin törəmələri polimorf hesab olunur. Bu o deməkdir ki, onların həm ümumi xüsusiyyətləri, həm də öz xüsusiyyətləri var. Məsələn nəslə ev olan qaraj nümunəsində polimorfizmin mümkünlüyünü izah edək: “Təsəvvür edək ki, qarajın qapısı açılır, evin qapısı isə yan tərəfə açılmalıdır. Ev qarajdandır, ona görə də onun qapısı da açılacaq. Onda necə olmaq olar? Yalnız evdə qapının açılması prosedurunu dəyişdirməlisiniz (yenidən yazmalısınız). Sonra ev qarajın bütün xüsusiyyətlərini alacaq, ancaq qapı açıldıqda, öz prosedurunu əvəz edəcəkdir. Başqa sözlə, polimorfizm müxtəlif iyerarxiyadakı obyektlərin eyni hadisəyə fərqli reaksiyasıdır. Polimorfizm əməliyyatlar və funksiyalar üçün çoxlu təriflər yaratmağa imkan verir. Hansı tərifin istifadə edilməsi proqramın kontekstindən asılıdır. OOP polimorfizmlə bağlı xüsusiyyətləri təqdim edir, məsələn: funksiya şablonları, funksiyaların həddən artıq yüklənməsi, operatorun həddən artıq yüklənməsi, virtual metodların istifadəsi, sinif şablonları (6, s. 163). Operatorun həddən artıq yüklənməsi sizə daxili C++ növlərində istifadə etdiyiniz eyni operatorları öz siniflərinizdə istifadə etməyə imkan verir. Virtual metodlar baza və törəmə siniflərin eyni adlı metodları arasında icra zamanı istədiyiniz metodu seçmək imkanı verir. Sinif şablonları məlumat növləri ilə bağlı invariant olan sinifləri təsvir etməyə imkan verir. İnkapsulyasiya “məyyyə bir tikinti blokundan (koddan) istifadə edən bir tərtibatçının bu tikinti blokundan düzgün istifadə etmək üçün əslində necə həyata keçirildiyini və işlədiyini bilməsi lazım olmayan bir xüsusiyyət” kimi başa düşülür.

OOP paradigmasının yaranması mürəkkəb proqram sistemlərinin qurulmasında yaranan problemlərin həlli zərurətindən irəli gəlirdi. OOP-un tarixi ilə bağlı ən mühüm hadisələr OOP-un əsas aspektlərini ehtiva edən Simula dilinin yaranması, həmçinin ilk obyektönlü proqramlaşdırma dili olan Smalltalk-ın meydana çıxmasıdır.

**Məqalənin aktuallığı.** OOP konsepsiyasının öyrənilməsinin aktuallığı ondan ibarətdir ki, OOP proqram təminatının işlənilməsində populyar proqramlaşdırma paradigmasıdır ki, bu da proqramçıya OOP-ni başa düşməyi və tətbiq etməyi zəruri edir.

**Məqalənin elmi yeniliyi.** Məqalədə obyektönlü proqramlaşdırmanın (OOP) əsas anlayışları və xüsusiyyətləri, habelə bu paradigmanın yaranması və inkişafı şəraiti müzakirə olunur.

**Məqalənin praktik əhəmiyyəti və tətbiqi.** Məqalədən informatika fənnini tədris edən müəllim və tələbələr istifadə edə bilər.

## Ədəbiyyat

1. Велов В.В. Основы языка Delphi. Программирование в среде Delphi. 2009.
2. Ащарина И.В. Объектно-ориентированное программирование в C++: лекции и упражнения.
3. Ə. Sadıqov. C++ proqramlaşdırma dili. Bakı, 2016.
4. İnternet resursu: [https://www.neansoft.com/kitab-borland\\_c\\_plus-plus\\_proqramlasdirma-dili.html](https://www.neansoft.com/kitab-borland_c_plus-plus_proqramlasdirma-dili.html).
5. S. Hüseynov. Obyektönlü proqramlaşdırma. Bakı, 2014.

**С.Дж. Джебраильзаде, Э.В. Исазде**

**Основные понятия и особенности  
объектно-ориентированного программирования**

**Резюме**

В статье рассматриваются основные понятия и особенности объектно-ориентированного программирования (ООП), а также условия возникновения и развития этой парадигмы.

**S.J. Jabrailzade, E.V. Isazade**

**Basic concepts and features of object-oriented programming**

**Summary**

The article discusses the basic concepts and features of object-oriented programming (OOP), as well as the conditions for the emergence and development of this paradigm.

**Redaksiyaya daxil olub: 05.05.2022**