

# Proqramlaşdırma Dillərinin Müasir Tədrisi Metodikasının Əsas Aspektləri

Fuad Qurbanov

Bakı Dövlət Universiteti, Bakı, Azərbaycan

*fndu@bk.ru*

**Xülasə—** Proqramlaşdırma dillərinin tədris metodikasının əsas aspektləri analiz edilmişdir. Tədris prosesinin effektivliyinin artırılmasının bir sıra yolları araşdırılmışdır. Proqramın keyfiyyətini və strukturunu müəyyən edən anlayışlar daxil edilmişdir.

**Açar sözlər—** *proqramlaşdırma, alqoritmik dil, səmərə, aspekt, tədris, sistem, mükəmməl kod, alqoritm, proqram gücü*

## I. GİRİŞ

Məlum olduğu kimi, müasir kompüter texnologiyalarının öyrənilməsinin və hazırlanma-sının əsas istiqamətlərindən birini proqramlaşdırma dilləri təşkil edir. Bu faktı onunla əsaslandırmaq olar ki, kompüterlərin fəaliyyətinin əsasını təşkil edən həm sistem, həm də tətbiqi proqram vasitələri bilavasitə proqramlaşdırma dillərinin köməyi ilə yaradılır.

Azərbaycanda yeni, yüksək səviyyəli və savadlı proqramçılar nəslinin yetişməsi ölkəmiz üçün son dərəcə aktual və vacib olan bir məsələdir. Bunun üçün respublikamızın ali məktəblərində və digər tədris ocaqlarında proqramlaşdırma dilləri və alqoritmlər nəzəriyyəsi fənlərinin tədrisini yüksək səviyyəyə qaldırmaq lazımdır.

Müasir zamanda cəmiyyətdə kompüter proqramlarına olan ehtiyac durmadan artmaqdadır. Tələb olunan proqram vasitələri getdikcə daha böyük həcmə və mürəkkəbliyə malik olurlar. Belə yüksək tələbatlara cavab verən proqramlarınaşağı səviyyədə təhsil almış mütəxəssislər tərəfindən hazırlanması mümkün deyildir. Bu isə öz növbəsində bir sıra müəssisələrin və şirkətlərin xaricdən olduqca yüksək qiymətlərə proqram vasitələri almasına gətirib çıxarır. Bütün bunların qarşısını almaq üçün ölkəmizdə günün tələblərinə cavab verən, beynəlxalq səviyyəli proqramçı mütəxəssislər yetişdirilməlidir. Bunun üçün isə tədrisin keyfiyyətinin artırılması yolları ciddi surətdə öyrənilməli və təcrübəli müəllimlərin uğur gətirən, müasir tələblərə cavab verən iş üsulları bütün tədris müəssisələrində istifadə olunmalıdır.

## II. TƏDRİS PROSESİNİN TƏŞKİLİ

Tədris prosesi məlum olduğu kimi mürəkkəb bir idarəetmə prosesidir. Burada insan amili böyük rol oynayır. Müəllimin hansı biliyə malik olması, tələbələrin qavrama səviyyəsi, tədris planı və s. amillər tədrisin keyfiyyətinə bilavasitə təsir göstərir.

Müəyyən bir prosesin keyfiyyətini yüksəltmək üçün təbii ki, onun modelini qurmaq, ona təsir göstərən amilləri

araşdırmaq və səmərəli inkişaf istiqamətlərini göstərmək lazımdır. Bu səbəbdən ən aşağı səviyyəli tədris prosesini “tədris oyunu”, ən yüksək səviyyəli təhsili isə “həqiqi təhsil” terminləri ilə adlandıracağıq. “Tədris oyunu” hər hansı bir proqramlaşdırma dilinin yalnız elementar hissələrinin tədris edilməsi və ya tələbələrdə alqoritmik dil haqqında səhv təsəvvürlərin yaradılması zamanı baş verir. “Həqiqi təhsil”-i isə biz dünyanın proqram mühəndisliyi sahəsində ən aparıcı təhsil ocaqları olan ABŞ-ın MIT, Kaliforniyanın Stenford, İngiltərənin Kembric, M.V.Lomonosov adına Moskva Dövlət Universiteti və digər bu kimi dünyanın yüksək səviyyəli təhsil ocaqlarında verilən təhsili başa düşürük. Bəs bu təhsillər arasındakı fərq nədən ibarətdir? “Tədris oyunundan” əsl “həqiqi təhsilə” hansı yollarla qalxmaq olar? Məqalənin əsas məqsədi bu yolların araşdırılması və bu istiqamətdə konkret fəaliyyət metodunun yaradılmasıdır. Hər şeydən əvvəl qeyd etmək lazımdır ki, hər bir real təhsildə bu iki komponentin hər biri müəyyən dərəcədə iştirak edir. Təhsilin idarə edilməsində birinci komponentin azaldılması, ikincinin isə artırılması yolları işlənib hazırlanmalıdır. Tədrisin əsl keyfiyyəti burada mövcud olan ikinci komponentin miqdarı ilə müəyyən edilə bilər.

Proqramlaşdırma sahəsində “həqiqi təhsil” dedikdə müəyyən bir proqramlaşdırma dilinin tam mənimsənilməsi, dilin çətin və abstrakt hissələrinin öyrənilməsi, onun köməyi ilə mürəkkəb alqoritmlər üçün böyük həcmli proqram tərtib etmə bacarığı başa düşülür. Başqa sözlə, “həqiqi təhsil” almış tələbə yüksək səviyyəli mütəxəssisə çevrilməlidir. “Təhsil oyunun”da isə əsl təhsil imitasiya edilir. Burada yalnız elementar səviyyəli, primitiv proqramlar tərtib edilir, yüksək, müasir biliklər öyrədilmir. Tədris rəhbərləri isə dərsi təşkil etdiklərini iddia etsələr də onun keyfiyyəti haqqında düşünmürlər. Belə təhsil prosesindən keçmiş tələbələr sonrakı fəaliyyətlərində böyük çətinliklərlə üzləşməli olurlar.

## III. PROQRAMLASHDIRMA TƏDRISİNİN ƏSAS ASPEKTLƏRİ

Proqramlaşdırma dillərinin tədrisi prosesi mürəkkəb bir sistem kimi təhlil edilə bilər [1], [2]. Burada təhsil prosesinə bilavasitə təsir edən əsas aspektlər müəyyən edilməli və onların tədrisin keyfiyyətinə göstərdikləri təsir imkanları araşdırılmalıdır. Ümumilikdə götürüldükdə tədris prosesinə təsir göstərən aşağıdakı əsas aspektləri göstərmək olar:

1. Proqramın mahiyyəti;
2. Proqrama qoyulan tələblər;

3. Alqoritm (riyaziaspekt);
4. Müasir texnologiyalar;
5. Tətbiq problemləri;
6. Psixoloji aspekt;
7. İntellektualaspekt;
8. Biliyin qiymətləndirilməsi;
9. Tələbənin fəaliyyət forması;
10. Yarışlara hazırlıq (olimpiadlar);
11. Mühazirə və seminarların forması;
12. Ev tapşırıqları;
13. Müasir ədəbiyyatın öyrənilməsi;
14. Tarixi aspektlər;
15. Dərsin strukturu;
16. İnformasiya mübadiləsi;
17. Müəllimin hazırlığı və fəaliyyət forması;
18. Texniki vasitələr.

Bu siyahıya bir sıra digər aspektləri də əlavə etmək olar. Ümumiyyətcə tədris prosesinə kompleksşəklində baxmaq lazımdır. Bu o deməkdir ki, yuxarıda göstərilən aspektlərin hər biri öyrənilməli və ayrıca bir istiqamət kimi inkişaf etdirilməlidir.

### 3.1. Proqramın mahiyyəti

Tədrisdə hər şeydən əvvəl proqramın mahiyyəti və tərkib hissələri,[3] araşdırılmalı və tələbədə bu bərdə səlis və aydın təsəvvürlər formalaşdırılmalıdır. Proqrama ilkin verilənlərdən yeni informasiya alan bir yaradıcı alət kimi baxmaq lazımdır [6]. Ümumiyyətlə, proqramı hansı dildə yazılmasından asılı olmayaraq aşağıdakı strukturda müəyyən etmək olar:

$$\text{Proqram} = \text{Alqoritm} + \text{Verilənlərin strukturu} + \text{Kod}$$

Tələbədə mükəmməl kod anlayışı yaratmaq və onu yüksək səviyyəli proqramlar yazmağa həvəsləndirmək əsas məsələlərdən biridir.

Mükəmməl kod [4, 5] dedikdə ən səmərəli üsullarla tərtib edilmiş, yığcam, optimal, kiçik zamanda düzgün nəticələr verən proqram başa düşülür. Bu anlayışı daha da dəqiqləşdirsək mükəmməl kodun aşağıdakı 3 əsas şərtini göstərə bilərik:

- düzgünlük;
- optimallıq;
- rahatlıq.

Düzgünlük dedikdə istifadəyə hazırlanmış proqramın nəzərdə tutulmuş çoxluqdan olan istənilən testə düzgün cavab verməsi başa düşülür. Əgər proqram hər hansı bir testə düzgün cavab vermirsə, demək o bu tələbatı ödəmir. Optimallıq dedikdə həm alqoritm, həm verilənlər strukturunun, həm də kodun optimallığı başa düşülür. Təbii ki, alqoritm optimallığı daha vacib sayılır. Optimal alqoritm ilə

yazılmayan proqram onun üçün ayrılmış zaman ərzində lazımi nəticəni verə bilməz. Proqramlaşdırma üzrə keçirilən olimpiadalarda alqoritmlərin optimallığına çox böyük əhəmiyyət verilir.

Mükəmməlliyin üçüncü şərti rahatlıq hesab edilir. Rahatlıq dedikdə proqram interfeysinə istifadəçi üçün tam rahat, cəlbedici və gözəl dizayna malik olması başa düşülür. Müasir proqramlaşdırma Delphi, Java, C# və s. kimi proqramlaşdırma dilləri mövcuddur. Bu dillər hər bir proqramçıya vizual vasitələrlə istifadəçi üçün yüksək səviyyədə rahatlığı təmin edən proqram vasitələri yaratmağa imkan verir.

Hər bir proqram müəyyən bir səviyyədə yazılırsə özündə müəllifinin bacarığını əks etdirir. Proqramların səviyyələrini əks etdirmək üçün “proqramın gücü” anlayışını aşağıdakı düsturun köməyi ilə daxil etmək olar:

$$\text{Proqramın gücü} = \alpha * (\text{Alqoritmın gücü}) + \beta * (\text{Verilənlər strukturunun gücü}) + \gamma * (\text{Proqram texnologiyalarının gücü})$$

burada,  $\alpha, \beta, \gamma$  əmsalları  $\alpha + \beta + \gamma = 1$  və  $0 \leq \alpha, \beta, \gamma \leq 1$  şərtlərini ödəyir.

Əmsalların qiymətləri proqramın tətbiq sahəsi, məqsədi və digər amillərlə müəyyən edilir. Təbii ki, proqramın ən əhəmiyyətli hissəsi alqoritm olduğundan, ümumi halda aşağıdakı bərabərsizliklər ödənməlidir:

$$\alpha \geq \gamma \geq \beta$$

### 3.2. Müasir texnologiyalar

Proqram üçün verilmiş düsturdan göründüyü kimi mükəmməl kodda verilənlərin strukturu da optimal seçilməlidir. Hər bir alqoritm daha səmərəli və tez işləməsi üçün ona uyğun xüsusi verilənlər strukturu yaradılmalıdır. Müasir kompüter elmində obyekt yönümlü proqramlaşdırmadan geniş istifadə edilir [9]. Sınıflar ən müasir strukturlar kimi artıq özlərini doğrultmuşlar. Vizual proqramlaşdırmada yeni komponentlərin siniflər vasitəsi ilə yaradılması müasir kompüter texnologiyalarının uğurlarından biri sayıla bilər.

Ümumiyyətlə, proqramlaşdırma texnologiyalarında bir sıra digər tələbatlar da mövcuddur. [7, 8]. Proqramçı təcrübəsini artırıqca yüksək səviyyəli mütəxəssisə çevrilməli və bu müasir tələbatları dərinlən mənimsəməlidir.

Tədris prosesi zamanı hər bir tələbə aşağıdakı müasir texnologiya komponentlərindən aktiv şəkildə istifadə etməlidir:

1. funksiyalar;
2. siniflər;
3. obyektlər və visual komponentlər;
4. dinamik yaddaş (Verilənlərin strukturu);
5. Cin rejimi (proqramı dayandırmadan müxtəlif verilənlər üçün nəticələrin alınması)
6. Verilənlərin üç cür daxil edilməsi:
  - a) ekrandan;

b) fayllardan;

c) təsadüfi seçimlə (random).

### 3.3. Alqoritm aspekti

Proqramlaşdırma tədrisinin ən əhəmiyyətli və vacib hissələrindən biri alqoritmlərin öyrədilməsidir. Proqramın fundamentini təşkil edən alqoritmlərin asimptotik qiymətləndirmə üsullarının mənimsənilməsi hər bir proqramçı üçün ən vacib keyfiyyətdir. Aydındır ki, belə qiymətləndirməni öyrənmədən heç bir mükəmməl kodun yaradılması mümkün deyildir [10].

Alqoritmik təfəkkürün inkişafı proqramçı üçün ən vacib məsələdir. Buraya müasir proqramlaşdırmada geniş tətbiq edilən alqoritmlərin və strukturların öyrənilməsi, alqotrimləri asimptotik qiymətləndirmə bacarığı, alqoritmləri mürəkkəblik dərəcələrinə görə siniflərə ayırmaq bacarığı və s. daxildir.

### 3.4 Psixoloji aspekt

Psixoloji aspekt dedikdə tələbənin proqramlaşdırmaya olan həvəsinin artırılması, öyrənmə prosesi zamanı meydana çıxan çətinliklərin öhdəsindən gəlmək üçün cəsarətin, çalışqanlıqın vəs. psixoloji amillərin formalaşdırılması başa düşülür. Burada tələbənin həvəsini artırmaq üçün nümunəvi proqramlar nümayiş olunmalı və məşhur proqramçılar barədə maraqlı məlumatlar verilməlidir. Maraqlı dairəsini artırmaq və gücləndirmək üçün əlavə dərslər, dərnəklər, klublar təşkil olunmalıdır. Yüksək nəticə əldə etməsi üçün tələbə bu fənni sevməli və vaxtının çox hissəsini maraqlı və çətin proqramların hazırlanması işinə həsr etməlidir.

### 2.5. Yarışlar

Proqramlaşdırma tədrisi zamanı ən vacib məqamlardan biri tələbələrin müxtəlif yarışlara, olimpiadalara hazırlanmasıdır. Olimpiadaların keçirilməsi tələbələrin bacarıqlarına obyektiv qiymətlər vermək üçün gözəl şərait yaradır. Öz səviyyələrini qaldırmaq üçün tələbələr internet saytlarında dərc edilmiş məsələlər üzərində çalışmalı və onların tam həllinə nail olmalıdırlar. Belə saytlardan aşağıdakıları misal göstərmək olar:

- ❖ Informatika.edu.az;
- ❖ E-olymp.com;
- ❖ Stepic.org;
- ❖ Codeforce.com;
- ❖ Topcoder.com;
- ❖ Neerc.ifmo.com.

Birinci üç saytlardan bütün tələbələrin, sonrakı üç saytdan isə daha yüksən səviyyəli tələbələrin istifadə etməsi məsləhət görülür.

## NƏTİCƏ

Yuxarıda göstərilən aspektlərin öyrənilməsi, tətbiqi və inkişafı proqramlaşdırma dillərinin tədrisinin keyfiyyətə daha yüksək səviyyəyə qalxmasına səbəb ola bilər. Hər bir aspektin həyata keçirilməsi müəllimin fərdi və subyektiv yanaşma tərzindən asılıdır. Bu səbəbdən təklif edilən üsullar müəllimlər tərəfindən öyrənilib tədris müəssisələrində istifadə edilə bilər. Təklif edilən metodlar və fikirlər çoxillik müəllimlik təcrübəsi zamanı əldə edilmişdir və müəllifin tədrisə yanaşma prinsiplərini özündə əks etdirir.

## ƏDƏBİYYAT

- [1] П.М. Хомяков, Системный анализ в 10 лекциях, Москва, АКИ, 2010, стр. 36-54.
- [2] R.M. Səmədəova, Ə.B. Məmmədov, “Sistemli yanaşma, sinergetik paradıqma”, Bakı, 2012, səh.163-209.
- [3] Н. Вирт, «Алгоритми +структура данных=программа», Москва, “Мир”, 1989, стр. 14-73.
- [4] Энди Орама, «Идеальный код», Москва, 2009, стр.46-60.
- [5] С. Макконелл, «Совершенный код», Москва, 2017, стр. 5-65.
- [6] А.Я.Фридланд, « Информатика, системы, ресурсы», Москва, 2003, стр. 29-39.
- [7] М.М.Бежанова, « Практическое программирование »,Москва, 2001, стр. 10-29.
- [8] К.И.Вигерс, «Разработка требований к программному обеспечению», Москва, 2004, стр. 2-72.
- [9] А.Л.Фридман, «Основы объектно-ориентированной разработки программных систем», Москва, Финансы и статистика, 2000, стр.5-52.
- [10] [Т. Кормен, «Алгоритмы Построение и анализ Москва», 2014, стр. 23-140.