



A.Z.FƏTƏLİYEVƏ

doktorant

e-mail: ayselfataliyeva33@gmail.com



:https://orcid.org/0009-0004-6279-777X

Azərbaycan Dövlət Pedaqoji Universiteti

(Bakı şəh., Ü.Hacıbəyov küç., 68)



10.62021/0026-0028.2025.2.291

ALİ TƏHSİLDƏ RƏQƏMSAL YARADICILIQ ÜZRƏ İNFORMATİKA VƏ PROQRAMLAŞDIRMA: FƏNNİNİN İNKİŞAF PERSPEKTİVLƏRİ

Xülasə. Bu məqalədə ali təhsil müəssisələrində “İnformatika və proqramlaşdırma” fənninin “Rəqəmsal yaradıcılıq” ixtisasları üzrə tədrisinin xüsusiyyətləri araşdırılır. Müxtəlif tədris yanaşmaları müqayisə edilərək, tələbələrin rəqəmsal texnologiyalar sahəsində praktiki bacarıqlarını inkişaf etdirmək üçün effektiv metod və vasitələr təhlil edilir. Python proqramlaşdırma dilinin tətbiqi, interaktiv və layihə əsaslı öyrənmə metodlarının üstünlükləri, eləcə də hackathon və dizayn yarışmalarının rolu vurğulanır. Məqalə, tələbələrin kreativ və analitik düşünmə bacarıqlarını artırmaq üçün müasir tədris yanaşmalarının əhəmiyyətini ortaya qoyur və ali təhsildə rəqəmsal yaradıcılıq yönümlü informatika tədrisinin inkişaf perspektivlərini müzakirə edir.

Açar sözlər: python, proqramlaşdırma, hackathon, rəqəmsal yaradıcılıq, alqoritmik düşüncə, rəqəmsal təhsil

Giriş. Müasir dövrdə informatika və proqramlaşdırma fənləri ali təhsil sistemində mühüm yer tutur. Bu sahənin tədrisi tələbələrin analitik düşünmə qabiliyyətini inkişaf etdirməklə yanaşı, onların texnoloji yeniliklərə adaptasiya olunmasını və müasir proqramlaşdırma vasitələri ilə effektiv işləməsini təmin edir. Xüsusilə Python proqramlaşdırma dili son illərdə ali təhsil müəssisələrində geniş tətbiq edilməyə başlanmış və müxtəlif sahələrdə – veri analitikası, maşın öyrənməsi, veb proqramlaşdırma və alqoritmik düşüncə inkişafı kimi istiqamətlərdə əsas vasitələrdən birinə çevrilmişdir.

İnformatika və proqramlaşdırma tədrisi sahəsində bir çox tədqiqatçı müxtəlif yanaşmalar irəli sürmüşdür. Wing “Computational Thinking” konsepsiyasını irəli sürərək proqramlaşdırmanın yalnız texniki ixtisaslar üçün deyil, ümumi məntiqi düşüncə və problem həll etmə bacarıqlarını inkişaf etdirmək üçün də önəmli olduğunu vurğulamışdır [8]. Grover və Pea isə informatikanın pedaqogikada tətbiqini araşdıraraq, tələbələrin proqramlaşdırma bacarıqlarını inkişaf etdirmək üçün interaktiv və layihə əsaslı tədris metodlarının daha effektiv olduğunu göstərmişlər [3].

Ali məktəblərdə Python dili ilə informatikanın tədrisi tələbələr üçün həm asan öyrənilə bilən sintaksisi, həm də geniş tətbiq sahələri baxımından xüsusi əhəmiyyət daşıyır. Python, tələbələrin proqramlaşdırmaya girişini asanlaşdırmaqla yanaşı, onların verilənlərin analizi, avtomatlaşdırma və süni intellekt sahələrində bilik və bacarıqlar əldə etməsinə şərait yaradır. Tədqiqatlar göstərir ki, ali təhsildə Python-un tədrisi zamanı problem əsaslı öyrənmə (PBL – Problem-Based Learning) və layihə əsaslı öyrənmə (Project-Based Learning) yanaşmaları daha uğurlu nəticələr verir [4]. Bundan əlavə, hackathonlar və interaktiv tapşırıqlar kimi fəaliyyətlərin informatika tədrisində effektivliyi son illərdə geniş müzakirə olunmuş və tələbələrin praktiki bacarıqlarını artırmaq üçün faydalı olduğu göstərilmişdir [1].

Bu məqalədə universitet tələbələri üçün informatika və proqramlaşdırma tədrisinin xüsusiyyətləri araşdırılaraq, Python-un ali təhsildə rolu, istifadə olunan metodologiyalar və praktiki tətbiqlər nəzərdən keçiriləcəkdir.



Metod. *İnformatika və proqramlaşdırmanın rəqəmsal yaradıcılıqda rolu, Rəqəmsal incəsənət sahəsində proqramlaşdırma bacarıqlarının əhəmiyyəti.* Rəqəmsal yaradıcılıq sahəsində proqramlaşdırma bacarıqları müasir incəsənət və dizaynın əsas elementlərindən birinə çevrilmişdir. Multimedia, animasiya və interaktiv incəsənət nümunələri proqramlaşdırma vasitəsilə yaradılır. Qrafik dizaynerlər və rəqəmsal incəsənət ustaları, alqoritmik düşüncə sayəsində daha yaradıcı və funksional əsərlər ortaya qoya bilirlər. Məsələn, kompüter qrafikası, interaktiv instalasiyalar, virtual və artırılmış reallıq layihələri proqramlaşdırma bilikləri olmadan effektiv şəkildə həyata keçirilə bilməz. Bu sahədə işləyən mütəxəssislər müxtəlif proqramlaşdırma dillərindən və alətlərdən istifadə edərək innovativ vizual layihələr icra edirlər.

Multimedia və interaktiv dizayn sahələrində tətbiq edilən proqramlaşdırma dilləri. Rəqəmsal incəsənət sahəsində proqramlaşdırma dilləri müxtəlif məqsədlər üçün istifadə olunur. Məsələn, JavaScript və Python interaktiv vizuallar və veb əsaslı animasiyalar yaratmaq üçün geniş istifadə edilir. Processing və OpenFrameworks kimi proqramlaşdırma mühitləri yaradıcı kodlaşdırma sahəsində mühüm yer tutur və vizual effektlərin generasiyasında geniş istifadə olunur. Həmçinin Unity və Unreal Engine kimi platformalar oyun dizaynı, 3D qrafika interaktiv simulyasiyalar üçün geniş tətbiq edilir. Bu alətlər rəqəmsal incəsənət mütəxəssislərinə kompleks multimedia layihələrini gerçəkləşdirmək üçün geniş imkanlar yaradır.

İnformatikanın incəsənət sahəsindəki praktik istifadəsi. İnformatika və proqramlaşdırma vasitəsilə rəqəmsal incəsənət layihələri daha interaktiv və dinamik hala gətirilir. Məsələn, generativ incəsənət layihələri məlumatların analizinə əsaslanan interaktiv vizuallar yaradır. İncəsənətçilər süni intellekt və maşın öyrənməsi alqoritmlərindən istifadə edərək yenilikçi rəqəmsal layihələr təqdim edirlər, yaradılan rəqəmsal incəsənət layihələri fərdi və interaktiv təcrübələr təqdim edir. Elektron sənət sərgilərində informatika vasitəsilə real vaxt rejimində dəyişən vizuallar və interaktiv sistemlər qurulur, virtual reallıq instalyasiyaları və istifadəçi ilə qarşılıqlı əlaqədə olan sistemlər qurulur.

Bundan əlavə, proqramlaşdırma qrafik dizayn və vizual effektlər üçün də geniş istifadə edilir. 3D modelləşdirmə, artırılmış reallıq və virtual reallıq tətbiqləri proqramlaşdırma biliklərinin istifadəsini tələb edir. Həmçinin HTML, CSS və JavaScript kimi veb texnologiyalar sayəsində veb əsaslı interaktiv incəsənət layihələri yaratmaq mümkündür.

Praktiki tapşırıqların rolu və əhəmiyyəti. *Hackathon və dizayn yarışmalarının tədrisdə rolu.* Hackathonlar və dizayn yarışları, tələbələrin yaradıcı düşünmə qabiliyyətini inkişaf etdirməklə yanaşı, onlara real problemlərini həll etmək, praktiki təcrübə qazanmaq və yeni texnologiyalarla tanış olmaq imkanı yaradır. Bu yarışlar, tələbələrin proqramlaşdırma, sistem analizi, dizayn və problem həll etmə kimi sahələrdə bacarıqlarını artıraraq, onları gələcək karyeralarına daha yaxşı hazırlayır [7]. Hackathonlar və dizayn yarışları, təhsil sistemində yalnız nəzəri bilikləri möhkəmləndirməklə kifayətlənmir, həm də tələbələrin praktiki bacarıqlarını inkişaf etdirmək üçün əvəzsiz bir vasitədir [6].

Hackathonlar və dizayn yarışları, tələbələrdən qısa müddət ərzində mürəkkəb problemləri həll etməyi tələb edir. Bu yarışlar, tələbələrin kreativ düşünmə qabiliyyətlərini inkişaf etdirərək, onlara problemləri daha geniş bir perspektivdən görə və müxtəlif həll yolları arasından ən uyğununu seçmə imkanı verir [5]. Bundan əlavə, bu yarışlar tələbələrin fərdi düşünmə bacarıqlarını gücləndirir və onlara komanda şəklində işləməyi öyrədir. Komanda üzvləri öz bilik və bacarıqlarını bölüşərək kollektiv düşünmədə effektivlik yaradırlar. Bu yanaşma, tələbələrin yalnız fərdi inkişafına deyil, həm də qrup işində effektivlik qazanmalarına kömək edir.

Bu yarışlar, tələbələrə praktiki bacarıqlar qazandırmaq üçün əvəzsiz bir fürsət təqdim edir. Məsələn, proqramlaşdırma dilləri, məlumat analitikası, süni intellekt və dizayn alətləri kimi sahələrdə biliklərini artırırırlar. Təhsil sistemində əsasən nəzəri biliklərə diqqət yetirilsə də, hackathonlar tələbələrə bu bilikləri real problemləri üzərində tətbiq etmə imkanı yaradır. Bu cür təcrübə, tələbələrin gələcək iş həyatlarında daha effektiv və səmərəli olmalarına kömək edir. Eyni zamanda, hackathonlar tələbələrə sürətli qərar qəbul etmə və təzyiq altında işləmə bacarıqlarını da öyrədir.



Azərbaycanda son illərdə keçirilən hackathon və texnoloji müsabiqələr, tələbələrin rəqəmsal texnologiyalar sahəsində marağını artırır. Bu yarışlar, tələbələrə yalnız yerli təcrübə qazanmağa deyil, həm də beynəlxalq səviyyədə öz bacarıqlarını sınamağa imkan verir. Məsələn, 2021-ci ildə “NASA Space Apps Challenge” yarışında Azərbaycanlı bir komanda iqlim dəyişikliyini monitorinq etmək üçün AI əsaslı bir alət təqdim etmişdir. Müsabiqələrdə iştirak edən tələbələr dünyanın müxtəlif ölkələrindən olan iştirakçılarla tanış olur, müxtəlif mədəniyyətlərdən və təcrübələrdən faydalanırlar. Bu onların beynəlxalq təcrübə qazanmalarına və global əmək bazarında öz yerlərini tapmalarına kömək edir. Eyni zamanda, bu yarışlar tələbələrə yeni texnologiyalarla tanış olmağa imkan verir. Məsələn, süni intellekt, maşın öyrənməsi, blokçeyn texnologiyaları və digər müasir texnologiyalar haqqında praktiki biliklər əldə edirlər. Bu cür təcrübələr tələbələrin gələcəkdəki karyeralarında onlara böyük üstünlük qazandırır.

Hackathonlar və dizayn yarışları, tələbələri yenilikçi fikirlər irəli sürməyə və tətbiq oluna bilən həll yolları tapmağa təşviq edir. Bu yarışların iştirakçıları, öz dizaynlarını və həll yollarını innovativ şəkildə təqdim edərək gələcək layihələr üçün ilham verə bilirlər. Azərbaycanda bu sahədə aktiv şəkildə iştirak edilir və müxtəlif təşkilatlar, universitetlər və şirkətlər bu cür yarışların keçirilməsini dəstəkləyir. Məsələn, Bakı Dövlət Universitetində 2022-ci ildə keçirilən bir hackathonda tələbələr kənd təsərrüfatında su idarəçiliyi üçün IoT əsaslı bir sistem hazırlamışdır. Bunun nəticəsində Azərbaycan tələbələri beynəlxalq səviyyədə tanınan texnologiya ekspertlərinə çevrilirlər.

Hackathonlar və dizayn yarışlarının tədris sahəsindəki rolu çox genişdir. Bu yarışlar tələbələrə yalnız nəzəri bilikləri deyil, həm də praktik bacarıqları inkişaf etdirməkdə yardımçı olur [2]. Onlar həmçinin tələbələrin zamanla təzyiq altında işləmək, komanda şəklində əməkdaşlıq etmək və müasir texnologiyalarla tanış olmaq qabiliyyətlərini artırır. Bu yarışların təsiri yalnız müvəqqəti deyil, həm də uzunmüddətli nəticələrə sahibdir. Tələbələr bu cür təcrübələr vasitəsilə öz güclü tərəflərini kəşf edir və özünü inkişaf etdirməyə davam edirlər. Hackathonların Azərbaycandakı inkişafı, ölkənin rəqəmsal texnologiyalar sahəsində daha güclü bir mövqeyə sahib olmasına kömək edir və tələbələrin beynəlxalq bazarda rəqabət qabiliyyətini artırır.

Praktiki tapşırıqların eksperiment hissəsi. *Hackathon və dizayn yarışmalarının tədrisdə rolu.* İnteraktiv proqramlaşdırma tapşırıqları tələbələrin öyrənmə prosesini daha maraqlı və effektiv edir, onların analitik düşünmə və yaradıcı problem həll etmə bacarıqlarını inkişaf etdirir [3]. Xüsusilə, vizual proqramlaşdırma mühitlərinin və layihə əsaslı yanaşmaların istifadəsi tələbələrin proqramlaşdırma biliklərini praktiki şəkildə mənimsəməsinə və yeni texnologiyalarla işləmə bacarıqlarını təkmilləşdirməsinə imkan yaradır [4]. Eyni zamanda, elektron tədris resursları və onlayn platformalar, məsələn, Coursera və Udey kimi sistemlər, tələbələrə fərdi öyrənmə imkanı verərək onlara müasir proqramlaşdırma yanaşmalarını daha dərinlən mənimsəməyə kömək edir [8]. Azərbaycanda ali təhsil müəssisələrində bu cür tədris metodlarının tətbiqi tələbələrin informasiya texnologiyaları sahəsində bilik və bacarıqlarını artırmaqla yanaşı, onları gələcək peşə fəaliyyətinə daha yaxşı hazırlayır [7].

Elektron tədris resurslarının və onlayn platformaların istifadə edərək tələbələr öz biliklərini genişləndirə bilirlər. Coursera, Udey və digər beynəlxalq platformalar vasitəsilə tələbələr interaktiv dərslərə qoşula bilirlər. Azərbaycan universitetlərində də e-tədris sistemlərinin inkişaf etdirilməsi və onlayn kursların yerli tədris prosesinə inteqrasiyası vacibdir.

İndi isə eksperiment hissəni verib ətraflı şərh edə bilərik.

Aşağıda verilən tapşırıqlar, tələbələrin Python proqramlaşdırma bacarıqlarını inkişaf etdirmək üçün klassikdən moderləşməyə doğru bir sıra tapşırıqlar təqdim edirik. Bu tapşırıqlar tələbələrin məntiqi düşüncə, alqoritmik bacarıqlar və real tapşırıqları həll etmə qabiliyyətlərini inkişaf etdirmək üçün tərtib olunub.

Klassik Tapşırıqlar (Əsas Alqoritmlər), proqramlaşdırmanın əsas prinsiplərini öyrənmək və inkişaf etdirmək üçün istifadə edilən tapşırıqlardır. Bu tapşırıqlar, yeni başlayanlara alqoritmlərin əsaslarını başa düşməyə kömək edir və proqramlaşdırma bacarıqlarını inkişaf et-



dirir. Bu növ tapşırıqların əsas məqsədi tələbələrə sadə riyazi əməliyyatları proqramlaşdırma vasitəsilə necə həll etməyi öyrətməkdir.

1. Sadə riyazi əməliyyatlar isə proqramlaşdırmanın əsas anlayışlarını öyrənmək üçün ən yaxşı başlanğıc nöqtəsidir. Riyazi əməliyyatlar, gündəlik həyatda tez-tez qarşılaşılan və insanın təcrübəsi ilə tanış olduğu tapşırıqlardır. Bu tapşırıqların həlli, həm alqoritmik düşünmənin inkişafına, həm də təməl proqramlaşdırma biliklərinin mənimsənilməsinə xidmət edir. Sadə Riyazi Əməliyyatlar tapşırığının nümunəsi, verilmiş iki ədədin cəmini, fərqlərini, hasilini və bölməsini hesablamaqdır. Bu tip tapşırıq proqramlaşdırmağa yeni başlayan şəxslər üçün ideal təlimdir, çünki burada əsas proqramlaşdırma elementləri, məsələn, dəyişənlər, operatorlar və sadə funksiyalar istifadə edilir.

1.1 Tapşırığın təsviri:

Verilmiş iki ədədin cəmini, fərqlərini, hasilini və bölməsini hesablayan proqram yazın.

python

Kopyala Düzenle

```
a = int(input("Birinci ədədi daxil edin: "))
b = int(input("İkinci ədədi daxil edin: "))

print(f"Cəm: {a + b}")
print(f"Fərq: {a - b}")
print(f"Hasıl: {a * b}")
print(f"Bölmə: {a / b if b != 0 else '0-a bölmək mümkün deyil!'})
```

1.2. Fibonacci sırası

Fibonacci Sırası proqramlaşdırmanın əsas anlayışlarını öyrənmək üçün faydalı bir tapşırıqdır. Fibonacci sırası, çox məşhur və təbiətdəki bir çox fenomenlə əlaqəli olan bir ardıcılıqdır. Bu tapşırıq, tələbələrə sadə alqoritmləri anlamaqda və proqramlaşdırma dilində təkrarlama (loop) və rekursiya kimi konseptləri öyrənməkdə kömək edir. Fibonacci sırası, ardıcıl olaraq iki ədədin cəmindən yaranan bir ardıcılıqdır: hər bir ədəd əvvəlki iki ədədin cəmidir. Məsələn, Fibonacci sırasının ilk bir neçə ədədi belədir: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

Fibonacci sırasının təsviri:

Fibonacci ardıcılığı, hər bir ədədin əvvəlki iki ədədin cəmi ilə müəyyən olunduğu bir sıra olduğu üçün, bu tapşırıq tələbələrə alqoritmik düşünməyi öyrədir. Fibonacci sırasının başlanğıcında iki ədəd, 0 və 1 yerləşir, bundan sonra ardıcıl olaraq hər bir ədəd əvvəlki iki ədədin cəmi kimi hesablanı bilər.

Verilən bir N ədədinə qədər olan Fibonacci sırasını tapmaq tələb olunur. Bu tapşırıq tələbələrə əsas operatorlardan, dövr strukturlarından (loop), və dəyişənlərin istifadə edilməsindən əlavə olaraq optimallaşdırma (məsələn, rekursiv və iterativ yanaşmalar) barədə anlayış verir.

Tapşırığı aşağıdakı kimi yazmaq olar.

N ədədinə qədər Fibonacci sırasını çap edən proqram yazın.

python

Kopyala Düzenle

```
n = int(input("Neçə Fibonacci ədədi çap edilsin? "))

a, b = 0, 1
for _ in range(n):
    print(a, end=" ")
    a, b = b, a + b
```



1.3. Sadə və mürəkkəb ədədləri tapmaq

Sadə və Mürəkkəb Ədədləri Tapmaq tapşırığı, tələbələrin proqramlaşdırma bacarıqlarını inkişaf etdirmək üçün əla bir təlimdir. Bu tapşırıq, alqoritmik düşünməyi gücləndirmək, şərt operatorlarından istifadə etməyi öyrətmək və dövr strukturlarını (loop) başa düşmək baxımından çox faydalıdır. Ayrıca, sadə (prime) və mürəkkəb (composite) ədədlərin fərqlərini öyrənmək də vacibdir.

Sadə Ədəd (Prime Number) Nədir?

Sadə ədəd (prime number), yalnız 1 və özündən başqa heç bir ədədə tam bölünməyən ədəddir. Başqa sözlə, sadə ədədin yalnız iki müsbət bölməni vardır: 1 və özünün. Məsələn:

- 2, 3, 5, 7, 11, 13, 17 və s. sadə ədədlərdir.

Mürəkkəb ədəd (Composite Number) nədir?

Mürəkkəb ədəd, 1 və özündən başqa ən azı bir digər ədədə tam bölünən ədəddir. Yəni bir mürəkkəb ədədin ən azı üç fərqli bölməni vardır: 1, özündən və bir və ya daha çox başqa ədəd. Məsələn:

- 4 (çünki $4 = 2 \times 2$), 6 (çünki $6 = 2 \times 3$), 8 (çünki $8 = 2 \times 4$) və s. mürəkkəb ədədlərdir.

Tapşırığın təsviri:

Verilmiş bir ədədin sadə olub-olmadığını yoxlamaq üçün proqram yazmaq tələb olunur. Əgər ədəd sadədirsə, proqram onu sadə olaraq çap etməlidir. Əgər mürəkkəbdirsə, proqram mürəkkəb olaraq göstərməlidir.

Həlli və yanaşma: Bu tapşırıqda, bir ədədin sadə olub-olmadığını müəyyən etmək üçün bölünmə prinsipindən istifadə olunur. Sadə ədəd, yalnız 1 və özünə bölünə bilən ədəddir, buna görə də verilmiş ədədin 2-dən həmin ədədə qədər olan ədədlərə bölünüb-bölünmədiyini yoxlamaq lazımdır.

Əgər ədəd heç bir ədədə tam bölünməzsə (yalnız 1 və özünə), o zaman bu ədəd sadədir. Əks halda, mürəkkəb ədəddir.

Tapşırığı aşağıdakı kimi yazmaq olar.

Verilmiş bir ədədin sadə olub-olmadığını yoxlayan proqram yazın.

```
python
def sadə_ədədmi(n):
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

n = int(input("Ədədi daxil edin: "))
print(f"{n} sadədir." if sadə_ədədmi(n) else {n} mürəkkəbdir.")
```

2. Orta səviyyəli tapşırıqlar (data strukturlar və alqoritmlər)

Bu tapşırıqlar tələbələrə real məsələlərə hazırlayır.

2.1. Siyahıda ən böyük və ən kiçik ədədi tapmaq

İstifadəçidən siyahı olaraq ən böyük və ən kiçik ədədi müəyyən edən proqram yazın.

```
python
nums = list(map(int, input("Ədədləri vergüllə ayıraraq daxil edin: ").split(',')))

print(f"Ən böyük ədəd: {max(nums)}")
print(f"Ən kiçik ədəd: {min(nums)}")
```



2.2. String əməliyyatları (Palindrom yoxlamaq)

Verilmiş bir sözün palindrom olub-olmadığını yoxlayan proqram yazın.

python

[Kopyala](#) [Düzenle](#)

```
s = input("Sözü daxil edin: ").lower()
print("Palindromdur" if s == s[::-1] else "Palindrom deyil")
```

2.3. Bubble sort ilə siyahını sıralamaq

İstifadəçinin daxil etdiyi ədədləri sıraya düzən Bubble Sort alqoritmini yazın.

python

[Kopyala](#) [Düzenle](#)

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]

nums = list(map(int, input("Ədədləri vergüllə ayıraraq daxil edin: ").split(',')))
bubble_sort(nums)
print("Sıralanmış siyahı:", nums)
```

3. Mürəkkəb və real tapşırıqları

Bu bölmə tələbələrin proqramlaşdırmanı real layihələrdə tətbiq etməsinə kömək edir.

3.1. Tələbə qiymət analizi (statistik hesablama)

Tələbələrin qiymətlərini qəbul edib orta balı, ən yüksək və ən aşağı nəticəni hesablayan proqram yazın.

python

[Kopyala](#) [Düzenle](#)

```
import statistics

qiymətlər = list(map(int, input("Qiymətləri vergüllə ayıraraq daxil edin: ").split(',')))

print(f"Orta bal: {statistics.mean(qiymətlər)}")
print(f"Ən yüksək bal: {max(qiymətlər)}")
print(f"Ən aşağı bal: {min(qiymətlər)}")
```



3.2. Şifrələmə (Caesar Cipher Algoritmi)

Mətnin hər hərfini müəyyən sayda dəyişərək şifrələyən proqram yazın.

```
python
def caesar_cipher(text, shift):
    result = ""
    for char in text:
        if char.isalpha():
            shift_amount = shift % 26
            new_char = chr(ord(char) + shift_amount)
            if char.islower() and new_char > 'z' or char.isupper() and new_char > 'Z':
                new_char = chr(ord(char) - (26 - shift_amount))
            result += new_char
        else:
            result += char
    return result

text = input("Şifrələnəcək mətni daxil edin: ")
shift = int(input("Şift dəyərini daxil edin: "))
print("Şifrələnmiş mətn:", caesar_cipher(text, shift))
```

3.3. Data vizualizasiya: imtahan nəticələrinin analizi

Tələbələrin imtahan nəticələrini göstərən qrafik hazırlayın.

```
python
import matplotlib.pyplot as plt

qiymətlər = list(map(int, input("Qiymətləri vergüllə ayıraraq daxil edin: ").split(',')))

plt.hist(qiymətlər, bins=5, color='blue', edgecolor='black', alpha=0.7)
plt.axvline(sum(qiymətlər)/len(qiymətlər), color='red', linestyle='dashed', label="Orta Bal")
plt.xlabel("Bal Aralığı")
plt.ylabel("Tələbə Sayı")
plt.title("İmtahan Nəticələrinin Paylanması")
plt.legend()
plt.show()
```

Bu hissədə aşağıdakı qənaətə gəlmək olar ki,

- **klassik tapşırıqlar** tələbələrə əsas alqoritmlərini qazandırır.
- **orta səviyyə tapşırıqlar** data strukturlar və alqoritmlərin real istifadəsini gücləndirir.
- **mürəkkəb tapşırıqlar** isə tələbələrin kodlaşdırma bacarıqlarını praktik yönümlü inkişaf etdirir.

İndi isə bu tapşırıqları Texniki universitetin kompüter elmləri, proqramyönümlü ixtisas tələbələrinin bacarıqlarını verilən misallar üzərində qrafik göstəricilərini əyin etmək olar.

Tələbə Nəticələrinin Təhlili və Qrafik Göstəricilər. Yazılacaq koda diqqət etsək, fərqli öyrənmə metodlarını tətbiq edən tələbələrin nəticələrini analiz edir və Boxplot (qutu qrafiki) ilə müqayisə edir.



Eksperimentin təsviri

Bu eksperiment, Azərbaycan Texniki Universitetinin 54 tələbəsi üzərində aparılmışdır. Araşdırmanın məqsədi, proqramlaşdırma bacarıqlarının inkişafını və tədris metodlarının effektivliyini qiymətləndirməkdir. Tələbələr, onların əvvəlki proqramlaşdırma təcrübələrinə və bacarıqlarına əsasən üç qrupa bölünmüşdür:

1-ci qrup (sadə tapşırıqlar, 18 nəfər): Bu qrupa daxil olan tələbələr sadə proqramlaşdırma tapşırıqları ilə məşğul olmuşlar. Bu tapşırıqların əsas məqsədi riyazi əməliyyatların və əsas alqoritmlərin tətbiqini nəzərdə tuturdu. Fibonacci sırası, müxtəlif riyazi əməliyyatların həlli və sadə alqoritmlərin yazılması tələb olunmuşdur. Bu qrupun məqsədi tələbələrə proqramlaşdırma dilinin əsaslarını və sadə alqoritmlərin qurulmasını öyrətmək idi. Tələbələrə verilən tapşırıqlar, onların məntiqi düşünmə bacarıqlarını inkişaf etdirmək və kodun sadə şəkildə yazılmasına kömək etmək üçün nəzərdə tutulmuşdur.

2-ci qrup (orta çətinlik, 18 nəfər): Bu qrupa daxil olan tələbələr daha mürəkkəb məsələlərlə üzləşiblər. Burada verilən tapşırıqların məqsədi, tələbələrin proqramlaşdırma biliklərini genişləndirmək və onlara müxtəlif data strukturlarını və alqoritmləri tətbiq etməyə öyrətmək idi. Bu tapşırıqlara misal olaraq, palindromların yoxlanması, siyahıların sıralanması və müxtəlif data strukturlarının istifadəsi verilə bilər. Bu qrupda iştirak edən tələbələrə, həmçinin real dünya problemləri ilə əlaqəli məsələlər təqdim edilib, onların proqramlaşdırma üzrə bacarıqlarının daha da təkmilləşdirilməsi məqsəd qoyulub.

3-cü qrup (mürəkkəb tapşırıqlar, 18 nəfər): Bu qrupa daxil olan tələbələr ən mürəkkəb və real dünya problemləri ilə məşğul olmuşlar. Onlar imtahan nəticələrinin statistik analizini aparmaq, şifrələmə alqoritmlərini təhlil etmək və həll etmək kimi yüksək səviyyəli tapşırıqları yerinə yetiriblər. Bu tapşırıqlar tələbələrə yalnız proqramlaşdırma bacarıqlarını deyil, həm də analitik düşünmə qabiliyyətlərini inkişaf etdirmək, real dünyada qarşılaşa biləcək problemləri həll etməyə yönəlmişdir. Bu qrupun məqsədi, tələbələrə müasir proqramlaşdırma metodlarını tətbiq etmə və daha mürəkkəb məsələləri həll etmə bacarığı qazandırmaqdır.

Araşdırma metodologiyası: Araşdırmanın aparılması metodoloji baxımdan çoxyönlü bir yanaşmanı əhatə edir. Hər bir qrup üçün müəyyən edilmiş tapşırıqlar tələbələrin fərqli proqramlaşdırma səviyyələrinə uyğun olaraq seçilmişdir. Bu metodologiya, müxtəlif çətinlik səviyyələrinə uyğun tapşırıqların tələbələrin nəzəri və praktiki biliklərini inkişaf etdirməyə kömək etdiyini göstərmək məqsədini güdür. Həmçinin hər bir qrupun nəticələri təhlil edilərək, proqramlaşdırma bacarıqlarının inkişafına təsir edən faktorlar araşdırılmışdır.

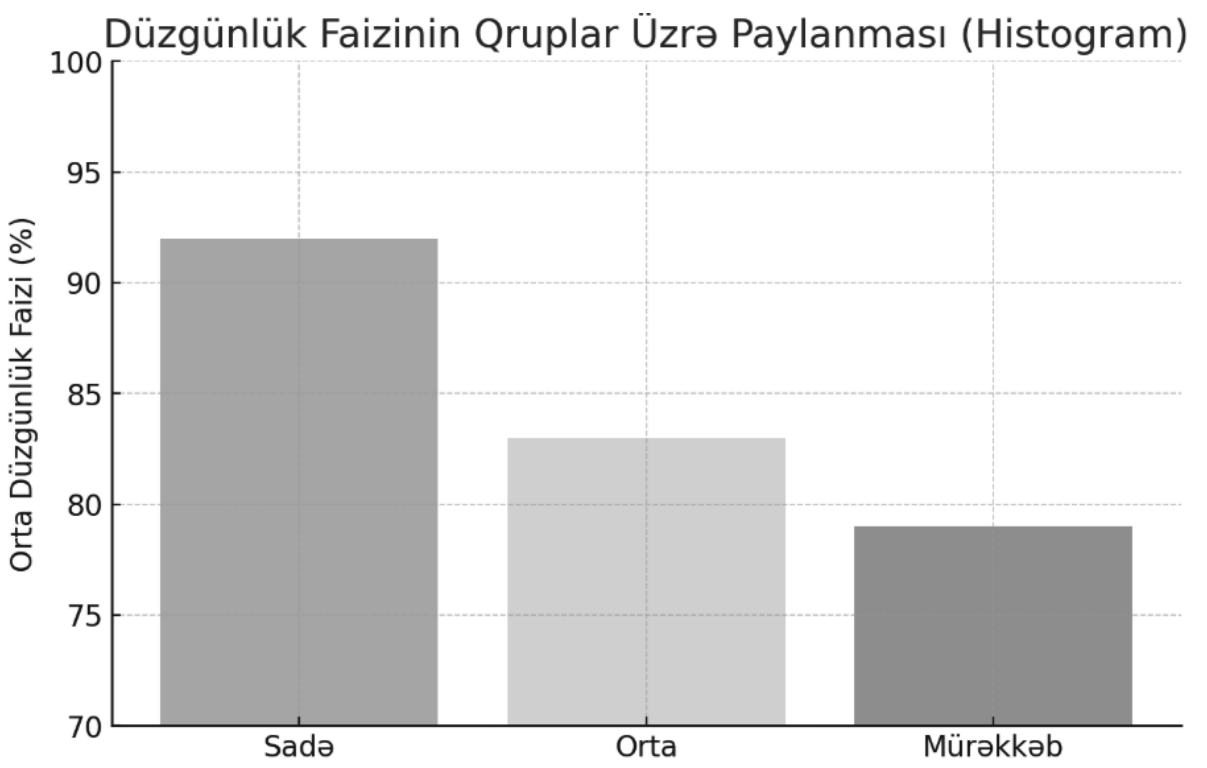
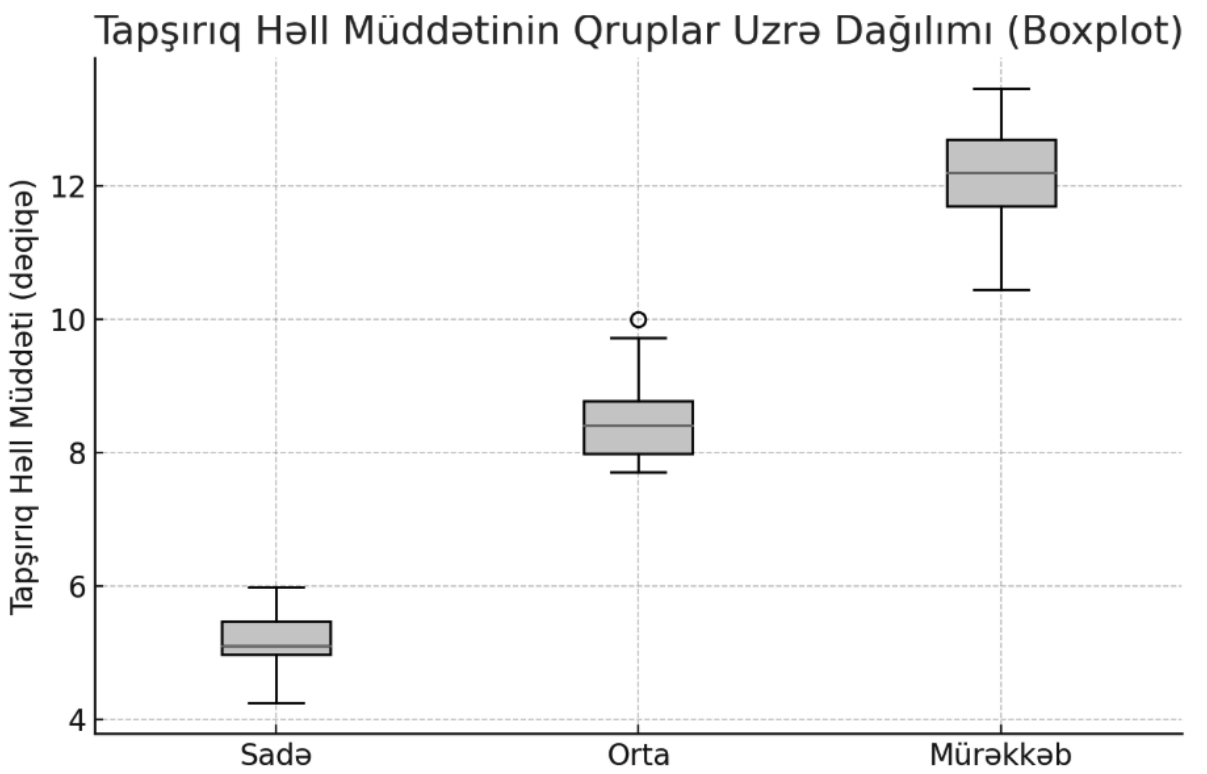
İstifadə olunan metodlardan biri, tələbələrin proqramlaşdırma bacarıqlarını ölçmək üçün obyektiv qiymətləndirmədir. Bu həm nəzəri biliklər, həm də tapşırıqların icrası üzrə təhlil edilən nəticələr əsasında həyata keçirilmişdir. Tələbələrin həll etdikləri tapşırıqların mürəkkəbliyi, həmçinin verilən tapşırıqlara tətbiq etdikləri alqoritmik yanaşmaların keyfiyyəti və optimallaşdırma səviyyəsi də qiymətləndirilmişdir.

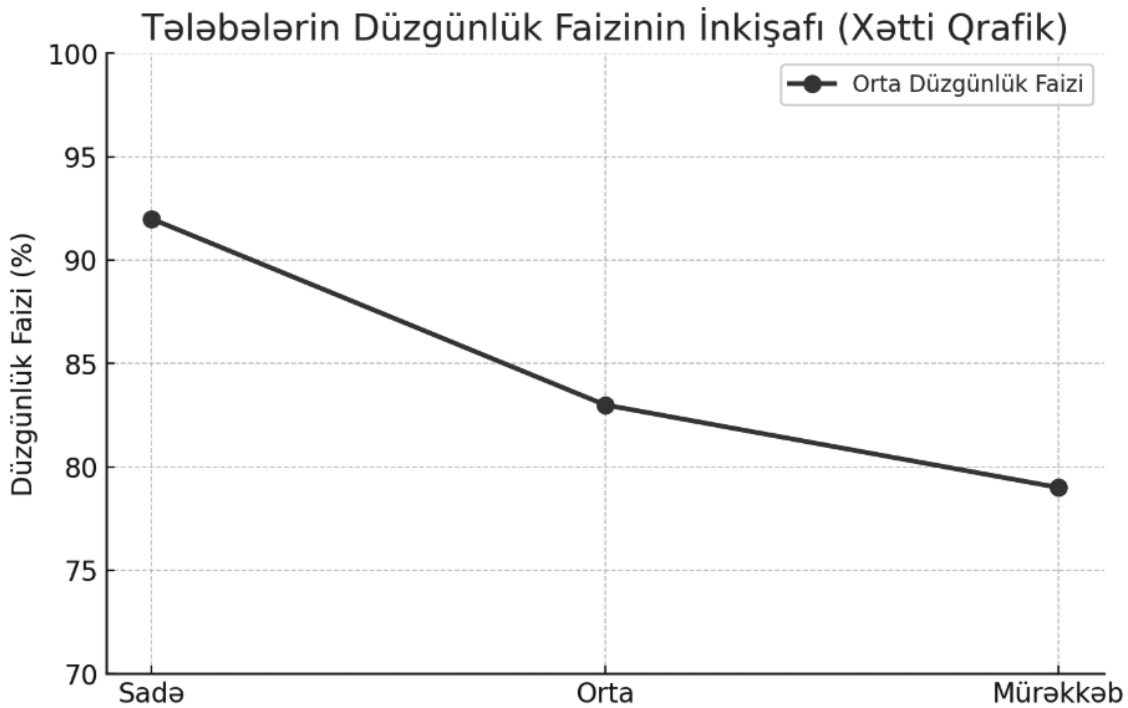
Nəticələrin təhlili, müxtəlif proqramlaşdırma biliklərinin tələbələrin fərqli qruplarda necə tətbiq olunduğunu, tələbələrin necə öyrəndiklərini və hansı çətinliklərlə qarşılaşdıklarını göstərir. Bu, tədris metodlarının təkmilləşdirilməsi və proqramlaşdırma dərslərində daha effektiv yanaşmaların tətbiq edilməsi üçün vacib məlumatlar verir.

Tələbələrin göstəriciləri aşağıdakı meyarlarla qiymətləndirilmişdir:

Qrup	Orta çətinlikli tapşırığın həlli müddəti (dəqiqə)	Orta düzgünlük faizi (%)
1-ci qrup (sadə)	5.2	92
2-ci qrup (orta)	8.7	83
3-cü qrup (mürəkkəb)	12.4	79

Qrafik olaraq meyarlara fikir versək, sadə dildə yuxarıda verilmişləri qrafiklərdə görə bilərik.





1. Boxplot – tapşırığın həll müddətinin yayılması.

Boxplot, verilən məlumatların yayılmasını vizuallaşdırmaq üçün istifadə edilən güclü bir alətdir. Bu qrafik, hər qrupun tapşırıq həllətmə müddətinin (dəqiqə ilə) necə paylandığını göstərir və tələbələrin performansını müqayisə etməyə kömək edir. Boxplotda aşağıdakı elementlər vizuallaşdırılır:

- **Median (orta dəyər):** Qrupun tapşırıq həllətmə müddətlərinin orta dəyərini göstərir. Median, məlumatların 50%-nin bu dəyərin altında, qalan 50%-nin isə bu dəyərin üstündə yerləşdiyini göstərir.

- **Minimum və maksimum dəyərlər:** Tələbələrin ən qısa və ən uzun tapşırıq həllətmə müddətləri göstərilir. Minimum, ən sürətli həlli göstərən tələbənin nəticəsini, maksimum isə ən uzun müddət sərf edən tələbəni təmsil edir.

- **İnterkvartil aralıq (IQR):** Qrup daxilində tapşırıq həllətmə müddətlərinin necə paylandığını göstərir. IQR, qrupun 25%-nin həll müddətlərini göstərən yuxarı və aşağı kvartil arasında olan məsafəni təmsil edir və məlumatların dağılmasının genişliyini əks etdirir.

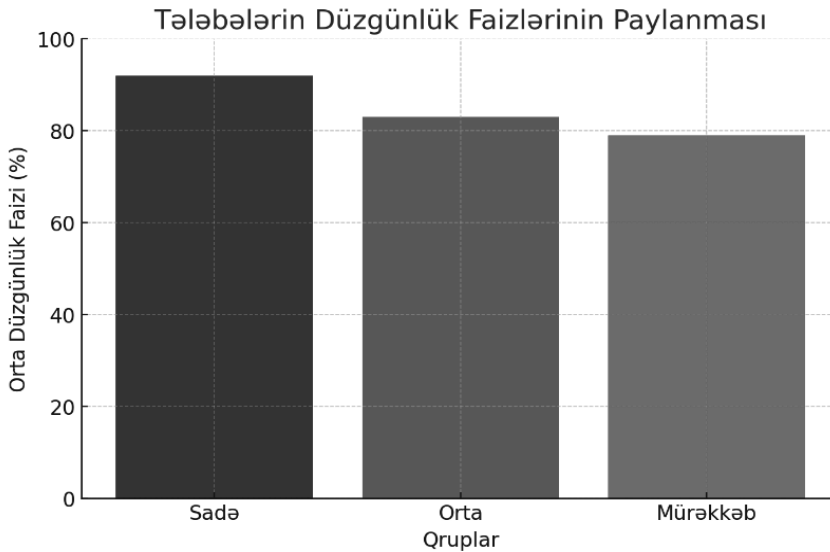
Bu qrafik, tapşırıq həllətmə müddətlərinin hər qrupda necə fərqləndiyini, qrup daxilindəki müxtəlifliyi və bəzi tələbələrin həll müddətindəki ekstremal fərqləri göstərmək üçün istifadə olunur. Məsələn, əgər bir qrupda həll müddətləri daha çox sıx və yığılmışsa, bu, tələbələrin tapşırıqları bənzər sürətlə həll etdiyini göstərə bilər, amma digər qrupda daha geniş yayılmış dəyərlər göstərsə, bu, performansın qeyri-bərabər olduğunu göstərir.

2. Histogram – düzgünlük faizinin paylanması

Histogram, tələbələrin düzgünlük faizlərinin paylanmasını əks etdirən bir qrafikdir və bu qrafik vasitəsilə hər qrupun düzgünlük faizlərinin necə bölüşdüyünü müqayisə etmək mümkündür. Histogram, hər bir qrupun düzgün cavablar ilə əlaqədar faizin hansı dərəcədə toplandığını göstərir və bu, ümumi performans barədə ətraflı məlumat verir.

Histogram qrafikini yaratmaq üçün verilənlərə əsasən verilənlər aşağıdakı kimidir:

Qrup	Orta düzgünlük faizi (%)
1-ci qrup (sadə)	92
2-ci qrup (orta)	83
3-cü qrup (mürəkkəb)	79



Tələbələrin düzgünlük faizlərinin paylanmasını göstərən histogram.

Qruplar arasında düzgünlük faizlərinin necə fərqləndiyini vizual olaraq müqayisə edə bilərik.

○ **X oxu:** Tələbələrin düzgünlük faizi ilə təmsil olunur (0%-dən 100%-ə qədər olan dəyərlər).

○ **Y oxu:** Müvafiq düzgünlük faizi olan tələbələrin sayı və ya faizi.

Bu qrafik, hər qrupun nə qədər yüksək düzgünlük faizi ilə nəticələndiyini göstərmək üçün istifadə edilir. Qrupun böyük bir hissəsi yüksək düzgünlük faizi göstərsə, bu, qrupun güclü performansını göstərir, əksinə, əgər düzgünlük faizi sıfır və ya çox aşağıdırsa, bu, tələbələrin tapşırıqları düzgün həll etməməsinin səbəblərini dərinləşdirmək üçün bir işarə ola bilər. Histogram, həmçinin hansı qrupun daha çox düzgün cavab verdiyini və hər bir qrupun performansını müqayisə etməkdə faydalıdır.

3. Xətti qrafik – düzgünlük faizinin inkişafı

Xətti qrafik, hər bir tələbənin düzgünlük faizlərinin zamanla necə dəyişdiyini və inkişaf etdiyini göstərir. Bu qrafik, tələbələrin performansını izləyərək, tədris müddətində onların düzgünlük faizindəki dəyişiklikləri analiz etməyə imkan verir.

○ **X oxu:** Vaxtı təmsil edir, məsələn, müxtəlif tapşırıqların yerinə yetirilməsi zamanı keçən zaman (tapşırıq nömrələri və ya tarixlər).

○ **Y oxu:** Hər tələbənin düzgünlük faizi.

Xətti qrafik, hər bir qrupun inkişafını göstərmək üçün istifadə edilir. Məsələn, bir qrupun düzgünlük faizi ilk tapşırıqlarda aşağı olsa da, zamanla artarsa, bu, tələbələrin öyrənmə prosesinin yaxşılaşdığını və tədris metodlarının onlara təsir etdiyini göstərə bilər. Həmçinin bu qrafik vasitəsilə qruplar arasındakı performans fərqləri göstərilə bilər, məsələn, bir qrupun düzgünlük faizi davamlı olaraq yüksək qalarkən, digər qrupun nəticələri dəyişkən və ya aşağı ola bilər. Bu cür təhlillər, tələbələrin hansı sahələrdə inkişaf etdiyini və hansı tapşırıqların onlara daha çox çətinlik yaratdığını daha dəqiq müəyyən etməyə kömək edir.

Bu üç qrafik birləşdirildikdə, tələbələrin səriştəlik bacarıqlarını daha ətraflı təhlil etmək, onların inkişafını izləmək və tədris metodlarını təkmilləşdirmək üçün güclü bir analitik alət təmin edir. Hər bir qrafik, fərqli aspektlərdən tələbələrin bacarıqlarını və onların zamanla necə inkişaf etdiyini göstərir.

Nəticə və metodik tövsiyələr: Məqalədə əldə edilən nəticələrə əsasən, rəqəmsal bacarıqlı tələbələr üçün proqramlaşdırma fənninin praktik yönümlü olması, interaktiv və layihə əsaslı metodların tətbiq edilməsi daha səmərəli olduğu qənaətinə gəlinir. Bu istiqamətdə ali məktəblərdə müasir texnologiyaların tətbiqinin artırılması və xüsusi tədris materiallarının hazırlanması tövsiyə olunur.



Ədəbiyyat

1. Ben-Ari, M. Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 17(1), 7-12. – 1998.
2. Flores, M., Gomes, J., Pereira, L., & Alves, P. Hackathons as Collaborative Learning Environments. *IEEE Global Engineering*. – 2018.
3. Grover, S., & Pea, R. Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. – 2013.
4. Guzdial, M. Does contextualized computing education help? *ACM Inroads*, 1(4), 4-6. – 2010.
5. Komssi, M., Pichlis, D., Raatikainen, M., Kindström, K., & Järvinen, J. What are Hackathons for? *IEEE Software*, 32(1), 60-67. DOI: 10.1109/MS.2014.100. – 2015.
6. Nandi, A., & Mandernach, M. Hackathons as an Informal Learning Platform. *ACM SIGCSE Technical Symposium*, 346-351. DOI: 10.1145/2839509.2844590. – 2016.
7. Porras, J., Knutas, A., Ikonen, J., Happonen, A., Khakurel, J., & Herala, A. Hackathons in Software Engineering Education: Lessons Learned from a Decade of Events. *Journal of Systems and Software*, 157, 110379. – 2019.
8. Wing, J. M. Computational thinking. *Communications of the ACM*, 49(3), 33-35. – 2006.

A.Z.Fətəliyeva

Development Prospects of Informatics and Programming for Digital Creativity in Higher Education *Summary*

This paper examines the teaching characteristics of the "Informatics and Programming" course in higher education institutions within the field of "Digital Creativity." By comparing various teaching approaches, the study analyzes effective methods and tools that enhance students' practical skills in digital technologies. The role of Python programming, the advantages of interactive and project-based learning methodologies, as well as the significance of hackathons and design competitions, are highlighted. The article emphasizes the importance of modern educational strategies in fostering students' creative and analytical thinking abilities and discusses the development prospects of digital creativity-oriented informatics education in higher education.

Key words: *Python, programming, hackathon, digital creativity, algorithmic thinking, digital education*

A.З.Фаталиева

Перспективы развития информатики и программирования в области цифрового творчества в высшем образовании *Резюме*

В данной статье рассматриваются особенности преподавания дисциплины «Информатика и программирование» в высших учебных заведениях в контексте специальности «Цифровое творчество». Сравниваются различные учебные подходы, анализируются эффективные методы и инструменты, способствующие развитию практических навыков студентов в области цифровых технологий. Подчеркивается роль языка программирования Python, преимущества интерактивного и проектного обучения, а также значение хакатонов и дизайнерских конкурсов. В статье акцентируется важность современных образовательных стратегий для развития креативного и аналитического мышления студентов, а также обсуждаются перспективы развития преподавания информатики, ориентированной на цифровое творчество в высшей школе.

Ключевые слова: *Python, программирование, хакатон, цифровое творчество, алгоритмическое мышление, цифровое образование*

Rəyçilər: p.ü.f.d. Ş.M.Ağazadə,
r.ü.f.d. G.A.Ağayeva

Redaksiyaya daxil olub: 11.03.2025