

Д.Р.Бейсенова, И.С.Қауымбек

Е.А.Бөкетов атындағы Қарағанды мемлекеттік университеті

ПӘНДЕР ИНТЕГРАЦИЯСЫ СТУДЕНТТЕРДІҢ АЛГОРИТМДІК ДАЯРЛЫҒЫН ЖЕТІЛДІРУДІҢ ДИДАКТИКАЛЫҚ ШАРТЫ РЕТІНДЕ

В статье показаны эффективность и важность интеграции предметов в совершенствовании алгоритмической подготовки студентов. Описана решающая роль математики в изучении языков программирования. Интеграция предметов в содержании различных предметов сформировывает основные термины, законы, теоретические понятия, осуществляя системность, непрерывность и взаимосвязь знаний, единые требования к общему методу через выполнение практических заданий и упражнений.

There were efficiency and importance of the integration of objects in improvement of the student's algorithmic training in the article. In the article role described in the study of programming languages. Integration of objects creates basic terms laws, theoretical concepts, realizes systematic, continuous, interdependence of the knowledge, united demands to the common method through execution practical tasks and exercises in the contents of different objects.

Пәнаралық байланыс пен бүгінгі педагогикалық сөздікте жиі айтылатын «интеграция» ұғымының мәніне тоқталайық. Пәнаралық байланыс дегеніміз — жалпы ғылымдар жүйесінің дидактикалық мақсаттарынан туындайтын оқу бағдарламаларының өзара шартты байланысы; оқытушы мен оқушы арасындағы іс-әрекеттер жүйесі; оқыту процесін жүзеге асырудың дидактикалық шарты; білім беру мазмұны ерекшелігінің бірі; оқушылардың білім деңгейін көтеру; жүйелі ойлауын дамыту, ғылыми-танымдық көзқарасын қалыптастыру процесін жүзеге асыратын дидактикалық шарт. Интеграция — ғылыми білім мен ғылымның тұтас жүйесін қалыптастыруда, бөлу, жіктеуді бірлікте көрсететін білімнің өзара ену, өзара сіңу процесі; тұтастықты қалыптастырып, білімді жүйелеу мен жинақтауда әр түрлі ғылымдарды біріктіріп, пәндердің өзара байланысын көрсететін пәнаралық байланыстың жоғары сапалық деңгейі. Интеграция — өзара органикалық байланыстағы білімдердің бір-бірімен кірігуі, оқушының ғылымды біртұтас түсінуіне қол жеткізуі. Бұдан шығатын қорытынды: пәнаралық байланыс оқыту процесінің мазмұндық-құрылымдық жүйесіне, сабақ аясына кіретін түсінік болса, ал интеграцияның ауқымы кең. Бүгінгі білім беру кеңістігінде интеграция ғылымдарды жақындастырады, соның негізінде интеграцияланған курстар, пәндер пайда болады, оқыту процесінің білім мазмұндағы үздіксіз сабақтастығы артады. Бірыңғай әлемдік білім кеңістігіне кіруге жол ашады, ғылым мен өндірістің кірігуіне, ғылым жетістіктерін өндіріске енгізуге жол ашады, яғни интеграциялық процесс пәнаралық байланысты тудырады және жүзеге асыруды қажет етеді. Оның төмендегідей тиімді әрі маңызды жақтары бар:

- жекелеген пәндерді ғылыми-танымдық мазмұнды ұқсас тақырыптарды бір мезгілде оқыту;
- әр түрлі пәндердің білім мазмұнында берілген барлық пәнге ортақ негізгі ұғымдарды, заңдарды, теориялық түсініктерді қалыптастыру;
- білімнің жүйелілігін, үздіксіздігін, сабақтастығын қамтамасыз ету;
- жаттығулар мен практикалық тапсырмаларды орындау арқылы жалпы әдістемеге қойылатын бірыңғай талаптарды жүзеге асыру;
- барлық пәнде оқытылатын кейбір тақырыптардың қайталануын жою;
- оқу процесінің білімдік, тәрбиелік, дамытушылық мақсатын, оқу мен тәжірибенің бірлігін жүзеге асыру.

Пәнаралық байланыс біржақты (бір пәнмен байланыс) және көпжақты (екіден көп пәндермен) болуы мүмкін. И.Д.Зверев, В.Н.Максимова [1] пәнаралық байланыстың негізгі үш типін бөліп көрсетті: мазмұндық-ақпараттық; операциялық-іс-әрекеттік; ұйымдастырушылық-әдістемелік. Бірінші тип мынадай түрлерден тұрады: ғылыми білім құрамы бойынша фактілік, ұғымдық, теориялық; сана туралы білімі бойынша философиялық, тарихи-ғылымилық, яғни гностикалық, семиотикалық, логикалық; құндылықтар бағдар білімі бойынша идеологиялық, яғни диалектикалық-материалистік, саяси-идеялық, этикалық, құқықтық. Екінші тип бойынша пәнаралық байланыстың төмендегідей түрлері ажыратылады: оқушыларда еңбек, құрылымдық-техникалық, есептеу-өлшемдік, эксперименттік, бейнелеушілік; сөйлеу іскерлігін қалыптастыруға мүмкіндік туғызатын теориялық білімді қолданудағы практикалық іс-әрекет тәсілдері бойынша практикалық; оқушылардың өздігінен білім әрекетін,

ұйымдастырушылық-танымдық, оқу, шығармашылық, ойлағыштық іскерлігін қалыптастыратын оқу-танымдық; іс-әрекет тәсілі бойынша жаңа білімді табудағы танымдық; бағалаушылық, қатынастық, шығармашылық-эстетикалық іс-әрекет іскерлігін дайындауға қажетті, құндылық-бағдарлық іс-әрекет әдісі бойынша құндылық-бағдарлық. Үшінші тип төмендегі түрлерді ажыратады: әр түрлі білім типтеріне байланысты меңгеру тәсілі бойынша репродуктивті, ізденушілік, шығармашылық, жүзеге асырылу уақыты бойынша курсаралық, циклішілік, цикларалық; жүзеге асырылу уақыты бойынша перспективті, ілеспелі, сабақтастық; пәндердің өзара байланыс әдісі бойынша біржақты, екіжақты, көпжақты; тұрақты жүзеге асу бойынша эпизодтық, тұрақты, жүйелі; оқу-тәрбие процесін ұйымдастыру деңгейі бойынша сабақтық, тақырыптық т.б.

Алгоритмдік даярлық студенттерде арнайы біліктілік пен дағдыны қалыптастыруға және дамытуға жағдай жасайды, олар алгоритм мәні мен қасиеттерін, алгоритмді жазу құралы ретінде программалау тілдерін, математика әдістері мен олардың қосымшаларының алгоритмдік сипатын түсінеді. Программалау тілдерін оқыған кезде кез келген программаны жазу үшін математикалық есептер беріледі. Студент программа жазу үшін математикалық есептің қалай шығарылатынын білуі керек. Студенттің алгоритмдік даярлығын қалыптастырып, жетілдіруде математика үлкен орын алады. Математика ғылымы өзінің қолданылуы жағынан ең жоғарыда тұрады. В.Ф.Одоевский: «Математика да, сөз ғылымдары да, заңдылық, саясат та философиялық тұрғыдан алғанда, Күннен сәуле алған планеталардай, бір-бірімен тығыз байланысты және олардың бір-біріне әсері күшті де болады», — деп атап өтті [2].

Мысалы, a, b ($a \leq b$) натурал сандары берілген. $a \leq p \leq b$ шартын қанағаттандыратын барлық p (жай сандар) сандарын алу керек.

Алдымен математикада жай сандар не екендігі жайлы анықтаманы еске салайық. Егер n ($n > 1$) натурал санының бір мен өзінен басқа бөлгіші болмаса, ондай сан жай сан деп аталады.

Ескерту. Жай сандарды іздегенде $[a; b]$ кесіндісінен тек тақ сандар мен 2-ні алу керек екені белгілі. $a = 1$ болған жағдайда анықтама бойынша 1 жай сан қатарына енбейді.

Программада қолданылатын шамалар:

n — жай сандыққа тексерілетін сан; *first* — бірінші тексерілетін сан; *simp* — логикалық типті шама; k — табылған жай сандар саны.

Есеп шығарудың негізгі кезеңдері:

1. a мен b мәндерін енгізу.
2. 2 санының $[a; b]$ кесіндісіне тиістілігін тексеру ($a = 1$ немесе $a = 2$ болғанда мүмкін).
3. *first* мәнін анықтау. Егер a жұп сан болса, онда $first = a + 1$, егер тақ болса және егер $a = 1$ $first = 3$, ал $a \geq 3$, $first = a$.
4. *first*-тан бастап b -ге дейінгі аралықтағы барлық тақ сандарды тексеру және экранға жай сандарын шығару (k айнымалысына олардың санын жинау).

Егер $k = 0$, онда берілген аралықта жай санның болмағаны.

```
#include <stdio.h>
#include <math.h>
void main(void)
{int a, b, n, first, k, i, simp;
a=2; b=1;
while (a>b)
    a=2; b=1;
    {printf("\na="); scanf("%d",&a);
    printf("\nb="); scanf("%d",&b);
    if(a>b) printf("\n a саны b санынан үлкен болмауы керек!");
    }
k=0;
printf("\n");
if((a==1)||(a==2)) {printf ("2"); k=1;}
if (a%2==0) first=a+1;
else if (a==1) first=3; else first=a;
for (n= first; n<=b; n+=2)
{simp=1;
```

```

i=3;
while ((i<=sqrt(n)) && (simp))
if (n%i==0) simp=0;
else i+=2;
if (simp) {printf (“%d,”n); k++;}
}
if (k==0) printf (“\nБұл аралықта жай сандар жоқ”)
}
Енді функция көмегімен программа жазсак:
#include <stdio.h>
#include <math.h>
int simple (int n);
{int i;
i=3;
while (i<=sqrt(n))
if (n%i==0) return 0;
else i+=2;
return 1;
}
void main (void)
{int a, b, n, first, k;
a=2; b=1;
while (a>b)
{printf(“\na=”); scanf(“%d”,&a);
printf(“\nb=”); scanf(“%d”,&b);
if(a>b) printf(“\na саны b санынан үлкен болмауы керек!”);
}
k=0;
printf(“\n”);
if((a==1)||(a==2)) {printf (“2”); k=1;}
if (a%2==0) first=a+1;
else if (a==1) first=3; else first=a;
for (n= first; n<=b; n+=2)
if(simple(n)) {printf(“%d”, n); k++;}
if (k==0) printf (“\nБұл аралықта жай сандар жоқ”)
}

```

Си программалау жүйесі Ассемблер, Паскаль, Фортран және Бейсик тілдеріндегі программалармен байланыс жүргізуге мүмкіндік береді. Си тіліндегі программаларда қолдану үшін Ассемблер, Паскаль және Фортран тілдерінде программа жазудың ережелері және де бұл тілдермен интерфейс ұйымдастыру үшін Си тілі элементтері қарастырылады.

Басқа тілдегі программа деп Си тіліндегі программдан шақырыла алатын Асемблердегі, Паскальдағы немесе Фортрандағы кез келген функция, подпрограмма немесе процедура аталады. Си-программдан басқа тілдегі программаны шақыру үшін, Си-дегі программа да, басқа тілдегі шақырылатын программа да байланыс және атаулар туралы келісілген түрде жүргізілу керек. Паскаль мен Фортран байланыс пен атаулар туралы бірдей келісімдерді қолданады. Олар Си тілінің байланыс және атаулар туралы келісімдерінен ерекшеленеді [3].

Келесі мысал бүтін сандар массивін өсу бойынша сұрыптауды көрсетеді. Сілтеме арқылы жіберілетін массивті сұрыптау үшін Паскаль функциясы (Фортран подпрограммасы) шақырылады.

Паскаль процедурасы немесе Фортран подпрограммасы қолданылған Си-программаның варианты:

```

#include <stdio.h>
int arr[20] = {12,1, 3, 24, 56, 0, 1, 6, 7, 23,
10, 9, 11, 25, 3, 5, 8, 15, 20, 33};

main ( )
{
int i;
void pascal srt(int near *);

```

```

printf ("Сұрыптауға дейінгі массив \n");
for (i=0; i < 20; i++)
    printf ("%d%c", arr [i], (i==19) ? '\n': ' ');
srt(arr);
printf ("Сұрыптаудан кейінгі массив\n");
for (i=0; i < 20; i++)
    printf ("%d%c", arr[i], (i==19) ? '\n': ' ');
}

```

Паскальдағы процедура:

```

module P;
    type arr= array [1 .. 20] of integer;
    procedure Srt (var a:arr) [public];
        var temp: integer;
    begin
        for var i:=1 to 20 do
            for var j:=i+1 to 20 do
                if a[i]>a[j] then
                    begin
                        temp:=a[i];
                        a[i]:=a[j];
                        a[j]:=temp;
                    end;
            end;
        end;
    end.

```

Фортрандағы подпрограмма:

```

SUBROUTINE SRT (A)
INTEGER [C] A [NEAR]
DIMENSION A(20)
INTEGER*2 I,J
INTEGER [C] TEMP
DO 20 I=1,20
    DO 10 J=I+1,20
        IF (A(I) .GT. A(J)) THEN
            TEMP = A(I)
            A(I) = A(J)
            A(J) = TEMP
        ENDIF
    10 CONTINUE
    20 CONTINUE
END

```

Сонымен, пәндер интеграциясының маңызын былай қорытамыз: оқу пәндері арасындағы өзара байланыстың болуы — ғылымдар негізін меңгерудің және білім жүйесінің дамуының қажетті шарты; дидактикалық көзқарастың қалыптасуы білім мазмұнының барлық құрамды бөліктерінің байланысын талап етеді; пәнаралық байланыс жан-жақты тәрбие жүйесінің барлық салаларын кешенді жүзеге асыруға ықпал жасайды; пәнаралық байланыс педагогикалық еңбектің ғылыми негізде тиімді ұйымдастырылуына көмектеседі; педагогикалық ұжымның барлық іс-әрекетінің бір-бірімен келісімді және демократиялық негізде жүріп отыруына әсер етеді.

Әдебиеттер тізімі

1. Зверев И.Д., Максимова В.Н. Межпредметные связи в современной школе. — М.: Педагогика, 1981. — 91 с.
2. Виргинский В.С., Одоевский В.Ф. Естественнонаучные взгляды. — М.: Наука, 1975. — 112 с.
3. Сағындықов К.М. Алгоритмдік тіл Си. — Қарағанды: ҚарМУ баспасы, 2002. — 306 б.