УДК 681.3.06

С.О. Асанова, К.Т. Абдыракманова
«Информатика» кафедрасынын ага окутуучулары,ОшТУ

WEB – ВИЗУАЛИЗАТОРЛОРДУН МЕКТЕП КУРСУНДА ПРОГРАММАЛООНУ КОЛДОНУЛУШУ

Макалада программист – үйрөтүүчүлөр гана эмес, атайын web – визуализаторлор (web – колдонуу жана алгоритмдин демонстрациялык иштөө процесси менен алектенген программист - үйрөтүүчүлөр) программалоону жогорку сапатта, тереңдетилген материал менен окутуусу каралган. Программалоону окутуунун негиздери проблемалык окутуу методикасына ылайык циклдик конструкциянын бөлүгү катары окутууну сунуштайт.

Ачкыч сөздөр: программалоо, үйрөтүүчү, web – визуализаторлор, проблемалык окутуу, окутуу методикасы.

С.О.Асанова, К.Т.Абдыракманова
старшие преподаватели кафедры «Информатика» ОшТУ

ПРИМЕНЕНИЕ WEB-ВИЗУАЛИЗАТОРОВ В ШКОЛЬНОМ КУРСЕ ПРОГРАММИРОВАНИЯ

В статье рассматривается метод повышения качества усвоения материала, который основан на использовании при обучении программированию обучающих программ-тренажеров, а именно web-визуализаторов (программы-тренажеры, являющиеся web-приложениями и демонстрирующие процесс работы алгоритмов). Предлагается обучение основам программирования, в частности циклических конструкций, используя методику проблемного обучения.

Ключевые слова: программирование, тренажер, web-визуализатор, проблемное обучение, методика обучения.

S.O. Asanova, K.T. Abdyrakmanova Senior lectures of computer science chair, OshTU

USE OF WEB-VISUALIZERS IN COURSE OF STUDIES PROGRAMMING

The article concerns the method based on the use of computer tutors in classroom instruction, namely web-visualizes (which are web applications demonstrating the way algorithms work). We suggest programming principles, specifically cyclic design using the method of problematical searching.

Key words: programming, web-visualizes, computer tutors, methods of teaching, problematical searching.

Мектепте программалоону окутуунун маанилүү өзгөчөлүгү жөнөкөй чыгарманын мүмкүнчүлүгүнө караганда, акылды талап кылуучу шаблондуу маселелерди кабыл алууну чектебейт. Фактылык жактан алсак, ар кандай реалдуу маселе программист аркылуу чечилип, стандарттуу эмес ойлоону жана иш-аракетти талап кылат. Чындыгында, мектепте программалоону окутуу элементардык конструкциядан башталат. Каалаган алгоритм

тилинде программалоонун негиздерин окуп үйрөнгөн соң типтүү маселелерди чыгара алат жана алгоритм маданиятында, ошондой эле мындан ары программалоону үйрөнүп кетүүдө таяныч пункт катары кызмат кылат. Маселелерди тандап алуу мектеп курсунун стандарттуу программасында көрсөтүлгөн. Окуучу программалоону жогорку деңгээлде өздөштүрүү менен алгоритмдик сызыктуу структурасында иштөөнү, тармактануу, тандоо, циклдер, программанын астында уюштуруп анын ичинде рекурсивдин жөнөкөй жана татаал типтерин: бүтүн сандык, материалдык, символдук, массивдик, жазуу сыяктууларды колдоно билет. Программалоону окутууда алгоритмдин төмөнкү түрлөрү аткарылат: сызыктуу алгоритмдин аткарылышы, шарттуу оператор, шарт алдындагы циклдер, шарттан кийинки циклдер, циклдер параметрлер менен, массивдин элементтерин суммалоо, массивде минималдуу жана максималдуу элементтерди издөө, массивдеги жөнөкөй сорттоо, сапты издөө, ЭЭМде көп нерсенин берилиши, рекурсияны уюштуруу, файлдан чыгуу, кирүү.

Информатиканын профилдик курсунда программмалоону үйрөнүүнү тереңдетип, төмөнкү темалар менен предметти кеңейтүүгө болот: дарак, даракты кыдыруу, ашыкча алгоритм, матрица, сан жана матрица менен иштөө, саптар, тизмелер, варианттар, индуктивдүү далилдер, капыстан улануучу генерациялар.

Бул учурда алгоритмдин тизмеси төмөнкүдөй алгоритмдер менен жайылат: матрицаны көбөйтүү, массивди жакшыртып сорттоо, сырткы сорттоо. Файлды сорттоо, кезек менен иштөө.

Мындай тизмеде жараксыз нерсе болбойт. Берилген маселени тандоо мектепте программалоону окутуунун максаты жана маселеси. Бир жагдайды эске алуу зарыл, эгерде окуучуга кандайдыр бир шаблонго байланышкан нерсе жеткиликсиз болсо, аны чыгармалык жактан өнүктүрүү мүмкүнчүлүгү жаралат. Программалоону окутууда атайын билим жана алгоритмдик ойлоо гана талап кылынбайт, алгоритмдөөнүн такай өнүгүү процесси, логикалык ойлоо да талап кылынат. Окуучунун логикалык ойлосуу мектептин баштапкы билим берүүсүндө математиканы окутуудагы логиканын негизинде эле башталат. Окуучунун логикалык ойлоо мүмкүнчүлүгү жаны билимди алууда активдүүлүгүнөн байкалат. Проблемалык окутуудагы логикалык ойлоону өз алдынча өнүктүрүүдөгү бирден бир эффективдүү метод окумуштуунун чыгармачыл иш-аракетине жакындоо, анын гипотеза, далил, эксперименттери менен таанышуу болуп эсептелет.

Проблемалык жагдай окуу материалынын конкреттүү мазмунунан көз каранды. Бир проблемалык жагдай окуучунун билимди терең алуусуна өбөлгө. Окуучу кебине каршы факт менен чыгууга аракет жасайт. Демек, кошумча маалымат алуу үчүн билим жетишсиз. Проблемалык кырдаалга мындай вариант окуучунун курч мүнөздө кызыгуусун жаратат, жогорку эффективдүүлүк менен таанышытырат.

Педагог жана психологдор мындай проблемалык окутуу методикасында логикалык ойлоону өнүктүрүүнү сунушташат.

Окуучу иш–аракет менен активдүү таанышуусу үчүн проблемалык кырдаал түзүүдө проблемалык методдун негиздери, татаал суроолордун чечимин табуу жана издөө, билимди актуалдуу талап кылуу, анализ, факт жана закондорду акыл менен байкоо сыяктуу нерселер эсептелет.

Мугалим проблемалык кырдаал түзүп, анын чечимин издөөгө окуучуларды уюштурат, багыттайт. Бул учурда бала жаны билимдин чечимин издөө, жаңы мүмкүнчүлүктү жаратууну көздөйт. Проблемалык кырдаал мугалим үчүн өзгөчө методикалык жол болуп эсептелет.

Мугалимдер проблемалык окутуунун мүнөзүнө жараша 4 деңгээлге болот:

1. Өз алдынча эмес активдүүлүк деңгээл – мугалимдин окуучуга түшүндүрүүсү, проблемалык кырдаал шартындагы акыл иш-аракетинин үлгүсү, окуучулар өз алдынча иш аткарат, көнүгүүлөр, оозеки чыгаруулар болуп эсептелет.

Өз алдынча эмес активдүүлүктө программадагы даяр тексттер үйрөтүлөт. Мисалы, циклдүү конструкцияны үйрөнүүдө – даяр алгоритмдин чечимин табуу, while циклинин жардамы менен 1ден N ге чейинки натуралдык сандардын суммасын табуудан башталат. Ал

үчүн таркатуучу материалдарды даярдоо, программа түзүү, берилген алгоритмди реалдаштыруу, комментарий жасоо:

```
        program sum1;
        var
        i, s, n: integer;
begin
        writeln('1денпгечейинкисандардыкийирүү');
        readln(n);
        i:=1; s:=0;
        while i<=n do begin
        s:=s+I;
        inc(i);
        end;
        writeln('s=',s);
end.
```

2. Жарым өз алдынча активдүүлүк деңгээлинде мугалим койгон суроо, проблема боюнча окуучулар катышып жообун издешет, жаңы кырдаалдарга даяр болот.

Бул деңгээлде жаңы материалды үйрөнүү мугалимдин айтып берүүсүнүн негизинде ишке ашат. Айтып берүүнүн мындай деңгээлдерде программадагы шаблондор колдонулат. Мисалы, окуучуга repeat циклин үйрөтүү үчүн 1ден N ге чейинки натуралдык сандардын суммасынын жардамы менен түшүндүрүлөт:

```
    program sum2;
    var
        i, s, n: integer;
begin
        writeln('1ден n ге чейинки сандарды кийирүү');
        readln(n);
        i:=1; s:=0;
        repeat
        ……
until
        …….
writeln('s=',s);
end.
```

Натыйжада мындай болот:

```
repeat
        s:=s+i;
        inc(i)
until i<=n;
```

3. Өз алдынча активдүүлүк деңгээли – репродуктивдүү – издөө тибинин аткарылышы, мында окуучу окуу китебиндеги тексти өзү түшүнө алат, мурдагы билимдерин эске алып жаңы кырдаалды кабыл алат, өзүн-өзү текшерет, татаал маселелерди орточо деңгээлде чече алат, мугалимдин жардамысыз гипотезаларды далилдейт жана башка.

Бул деңгээлде окуучуларга суроо берип, аны мугалимдин жардамысыз түшүнө алышат. Мисалы: 1ден Nге чейинки натуралдык сандардын суммасын for циклинин жардамы менен чыгарабыз:

```
    program sum3;
    var
        i, s, n: integer;
begin
```

```
writeln(‘1денпгечейинкисандардыкийирүү’);
readln(n);
for i:=1 to n do
s:=s+i;
writeln(‘s=’,s);
end.
```

Өз алдынча тапшырманын сапаты – факторду саноо маселеси (1ден Nге чейинки натуралдык сандардын чыгарылышы).

4. Чыгармачыл активдүүлүк деңгээли – өз алдынча иштерди аткаруу, чыгармачыл элестөөнү талап кылуу, логикалык анализ жана божомол, окуу проблемасында маселени чыгаруунун жаңы мүмкүнчүлүгү, өз алдынча далилдер, өз алдынча чыгаруу жана жалпылоо, ойлоп чыгаруу, көркөм дилбаяндарды жазуу.

Бул деңгээлде маселеге суроо коюш керек, анын чечимин табууда окуучу өз алдынча жөн эле чыгарбай, чыгармачылык менен иштейт. Проектте методдун элементтердин чыгармачылык менен колдонуучуга стимул болот. Окуучу өз алдынча формулалоо жана циклдик конструкцияны колдонуу менен практикада чектелген маселелерди чыгара алат. Мисалы, окуучу программа жазганынбелгилеп, алпортфелдегидептердинсанынорточочыгара ала алат:

```
program notebook;
var i, k, s:integer;
p:real;
begin
writeln(‘окуучулардын санын кийирүү’);
readln(n);
s:=0;
for i:=1 to n do begin
writeln(‘дептерлердин санын кийирүүу , i, ‘окуучулар ’);
read(k);
s:=s+k;
end;
p:=s/k;
writeln(‘дептерлердин орточо саны’, p);
end.
```

Окуучу маселени чыгарат, демек, ал өз алдынча математикалык модель түзө алат. Биринчи кадамда дептерлердин жалпы санынын суммасын ар бир окуучуга маалымдайт. Суммалоонун аягында саналган маанилер окуучулардын санына бөлүнөт.

Аналогиялык мисалдар бардык курстун окуусунда колдонулат. Окуучулар менен жекече иш алып барууда кыйынчылык жаралбайт. Чындыгында, методиканын эффективдүүлүгү ар бир окуучуга жетет, ал маселелерди өзү чыгара алат. Тилекке каршы, мугалим ар бир окуучуга көңүл буруусу кыйын. Ал үчүн визуализаторлор (программа-үйрөтүүчүлөр) алгоритмдин ички логикасын демонстрациялап, окуучулардын иш-аракетин башкарат. Интернеттеги байланыш аркылуу үйрөтүүчүлөрмаанилүү мүмкүнчүлүктү жарата алышат.

Мындай технологияда түзүлгөн нерселер (технологиялык бул үйрөтүүчүлөрдүн web-мисалда түзүлгөн түрү) жалпы камсыздалат. Программист – үйрөтүүчүлөр web-мисалдар жана алгоритмдин демонстрациялык иштөө процессин жакшыртат, аларды web-визуализаторлор деп айтабыз. Мындай өндүрүлгөн нерселер салттуу класстык – сабак системасында жана өз алдынча иштөөдө да кеңири кабыл алынган.

Адабияттар:

1. **Калдыбаев, С.К.** Компьютердик окутуу программалары-программалап окутуунун заманбап багыты С.К.Калдыбаев, Т.А.Курамаева // Ж.Баласагын атындагы КУУнун жарчысы.- Бишкек, 2012

2. **Бердиев, А.** Компьютер: колдонуу, программалоо. (DOS, Basic, Turbo Pascal) А.Исраилов, Р.Табышов, Э.Осоров, Бишкек 2010
3. **Омаралиев, А.** «Информатиканыннегиздери» Р.ТабышовБишкек 2003
4. **Климова, Л.М.** Pascak 7.0. Москва2003
5. **Маренко, А.И.** Программирование в среде Турбо Паскаль 7.0. Киев 2003