УДК 372:378

Т. А. Присяжнюк,
аспірант
(Житомирський державний університет імені Івана Франка)

ЗАСТОСУВАННЯ ЕЛЕМЕНТІВ СТРУКТУРНО-ЛОГІЧНОГО МИСЛЕННЯ ДО РОЗВ'ЯЗАННЯ ЗАДАЧ ЗАСОБАМИ НИЗХІДНОГО ПРОЕКТУВАННЯ

Визначено та проаналізовано основні поняття структурного програмування. Пропонуються шляхи ефективного застосування методу структурного програмування у професійній діяльності майбутнього вчителя інформатики, застосовано елементи структурно-логічного мислення до розв'язання задач засобами низхідного проектування. Наведено приклади застосування методу низхідного проектування до розв'язування задач з шкільної інформатики.

На сучасному етапі розвитку суспільства, етапі становлення та розбудови нової системи освіти і науки в Україні, входження української освіти в Болонський процес, все більшого значення набуває науковий пошук нових, досконаліших методів роботи в навчанні та вихованні майбутніх фахівців, які б поєднували сучасні інформаційно-комунікаційні засоби та технології навчання з особистісним розвитком суб'єктів навчання. Постала нагальна проблема реформування національної системи освіти, яка спрямована на інформатизацію, автоматизацію та використання нових педагогічних технологій навчання, що забезпечать доступ до актуальних знань, підтримки прагнень до неперервної освіти і можливостей найповнішої самореалізації, формування компетентної особистості, а також передумов професійного зростання і мобільності в умовах сучасного суспільства.

Як зазначено в нормативних документах, одним із головних шляхів розв'язання даної проблеми є підвищення творчої активності майбутнього вчителя у процесі його професійно-педагогічної підготовки [1], зокрема вчителя інформатики, до розвитку структурно-логічного мислення старшокласників, оскільки це підвищить питому вагу готової інформації, змінить співвідношення між структурними елементами змісту освіти на користь засвоєння учнями способів пізнання, набуття особистого досвіду творчої діяльності.

Одним із шляхів вирішення даної проблеми може бути виокремлення структурного програмування як методології, яку запропонував Е. Дейкстра [2; 3] та доповнив Н. Вірт [4]. Незалежно один від одного вони виступили за відмову від оператора *goto* у програмуванні та вдосконалення *теорії мови програмування* загального використання. У результаті цього була запропонована методика *покрокової розробки програм* – від глобального до локального, від загального до часткового, тобто занурення в *алгоритм зверху донизу*. Незважаючи на значну зацікавленість та різні підходи до вдосконалення структурного програмування [5; 6], ця проблема залишається в полі зору науковців та є недостатньо розробленою, що і спричинило ідею написання даної статті.

Удосконалення методики розв'язання алгоритмічних задач на підставі ідей структурного програмування є головною метою нашої статті.

При побудові алгоритму виникає необхідність пояснення деяких складних дій, елементарними прикладами яких можуть бути ситуації з життя. Ставлячи вперше перед дитиною завдання прибрати в кімнаті, необхідно пояснити, як витирати пил, тримати віник у руках, як набирати воду в миску (чи у відро), як полоскати ганчірку та мити підлогу. Надалі такі пояснення будуть зайві, оскільки алгоритм "прибирання в кімнаті" вже відомий для дитини.

У результаті чого можна зробити висновок, що кожна задача є окремою командою для виконавця, якщо він знає алгоритм її виконання. В іншому випадку, виникає потреба розкласти задачу на так звані "підзадачі", які є "посильними" для виконання. Застосовуючи цей метод, отримують алгоритм, що складається з простих команд, які зрозумілі виконавцю, або розуміють, що дана задача є непосильною для вибраного виконавця. Прикладом цього може бути деталізація алгоритму побудови літака для дитини, в результаті якої виявляється, що задача є непосильною для виконання.

Запропонований підхід до конструювання алгоритмів називається *методом покрокової деталізації зверху донизу*. При цьому кожна операція буде подана у вигляді лише одного з трьох типів базових структур алгоритмів – *лінійної* (слідування – операції виконуються послідовно одна за одною – рис. 1), *розгалуження (*виконується відповідна операція в залежності від умови – рис. 2*)* та *повторення* (цикл – багаторазове повторення певної дії – рис. 3). Степінь деталізації алгоритму в даному випадку залежить від можливостей обраного виконавця.

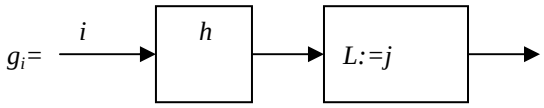

Рис. 1

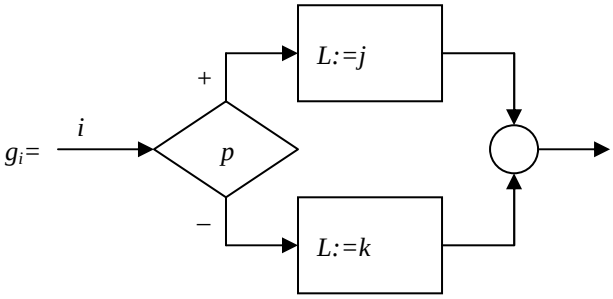

рис. 2

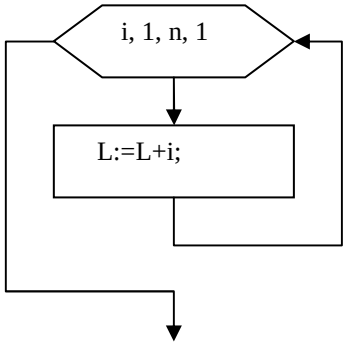

Рис. 3

Розглянемо найпростіший алгоритм переходу людини через вулицю. Для кожної дитини ця команда вважається відомою з дитинства, оскільки батьки неодноразово роз'яснювали правила дорожнього руху, які стали для дитини певним алгоритмом: якщо необхідно перейти вулицю, потрібно відшукати світлофор і скористатися основними правилами переходу за допомогою світлофора; якщо за даних умов світлофору не виявиться – правилами переходу без світлофора. При цьому в обох випадках виникає потреба деталізувати даний алгоритм – роз'яснити основні правила переходу за допомогою світлофора та без нього.

Проте існує ряд інших випадків, коли дана ситуація постійно доповнюється нескінченною послідовністю незрозумілих питань та вимагає застосування підзадач алгоритму, які називаються допоміжними. Вони створюються при поділі складної задачі на прості або при необхідності багаторазового використання одного й того ж набору дій в одному або різних алгоритмах.

Описаний метод послідовної деталізації лежить в основі технології структурного програмування й широко застосовується при використанні таких мов програмування, як Паскаль, С, С++ та інших.

У ході опису програми для комп'ютера мовами високого рівня допоміжні алгоритми реалізовуються у вигляді підпрограм. Правила описання, звернення до яких та повернення в точку виклику, визначаються конкретною мовою програмування. Для зручності часто використовувані підпрограми можна об'єднувати в бібліотечні модулі та при необхідності підключати їх у свої програми.

Розглянемо приклад розв'язання задачі з математики за допомогою вищевказаного методу.

Задача. Розв'язати біквадратне рівняння $ax^4 + bx^2 + c = 0$.

Математичний розв'язок.

Вводимо заміну змінної $y = x^2$ і отримуємо звичайне квадратне рівняння $ay^2 + by + c = 0$.

У залежності від коефіцієнтів a, b, c рівняння може бути різного типу.

Зокрема, якщо $a=0$, то рівняння буде лінійним $by+c=0$, причому:

(початок першого блоку)

при $b=0,\ c=0$ розв'язком буде множина усіх дійсних чисел;

при $b=0, c\neq 0$ розв'язків не буде;

при $b\neq 0, c\neq 0$ розв'язок буде один: $y=-c/b$.

У залежності від того, яким цей розв'язок буде (додатнім, від'ємним, чи рівним нулю), рівняння відповідно буде мати 2 розв'язки, один або жодного:

якщо $y>0$, то $x_1=\sqrt{y},\ x_2=-\sqrt{y}$;

при $y=0$ маємо $x=0$;

якщо $y<0$, то розв'язків немає.

(кінець першого блоку)

Якщо ж $a\neq 0$ (початок другого блоку), то знаходимо дискримінант за відомою формулою $D=b^2-4ac$. Тепер все залежить від дискримінанта. Якщо $D<0$, то рівняння розв'язків немає, інакше при $D=0$ (початок третього блоку), то рівняння має два рівних розв'язки, які можна обчислити за формулою $y_{1,2}=-\frac{b}{2a}$. При чому якщо $y>0$, то $x_1=\sqrt{y}$, $x_2=-\sqrt{y}$, при $y=0$, то $x=0$, інакше, тобто $y<0$ – розв'язків немає (кінець третього блоку). Якщо ж не виконується жодна з попередніх умов для дискримінанта, тобто $D>0$ (початок четвертого блоку), то рівняння має два різних розв'язки, і вони обчислюються за формулами $y_{1,2}=\frac{-b\pm\sqrt{D}}{2a}$. Тепер у залежності від отриманих розв'язків, рівняння буде мати 2, 3, 4 або жодного розв'язку. Розглянемо кожен випадок. 1) $y_1<0, y_2<0$ – розв'язків немає, 2) $y_1>0, y_2<0$ – будемо мати два розв'язки $x_1=\sqrt{y_1}, x_2=-\sqrt{y_1}$ 3) $y_1<0, y_2>0$ – буде також два розв'язки $x_1=\sqrt{y_2}, x_2=-\sqrt{y_2}$, 4) $y_1<0, y_2=0$ або $y_1=0, y_2<0$ – буде один розв'язок $x=0$, 5) $y_1=0, y_2>0$ – маємо три розв'язки $x_1=\sqrt{y_2}, x_2=-\sqrt{y_2}, x_3=0$, 6) $y_1>0, y_2=0$, (початок п'ятого блоку) – буде також три розв'язки $x_1=\sqrt{y_1}, x_2=-\sqrt{y_1}, x_3=0$, і 7) $y_1>0, y_2>0$, то буде чотири розв'язки $x_1=\sqrt{y_2}, x_2=-\sqrt{y_2}, x_3=\sqrt{y_1}, x_4=-\sqrt{y_1}$ (кінець п'ятого блоку) (кінець четвертого блоку) (кінець другого блоку).

Покажемо тепер цей розв'язок за допомогою схеми алгоритму (рис. 4).

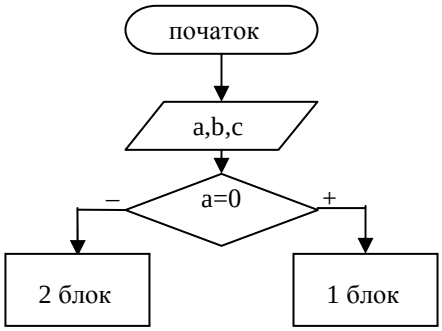

Рис. 4

Задача звелась до двох простіших. Дослідимо окремо кожну з них. Для цього розглянемо перший блок (рис. 5).

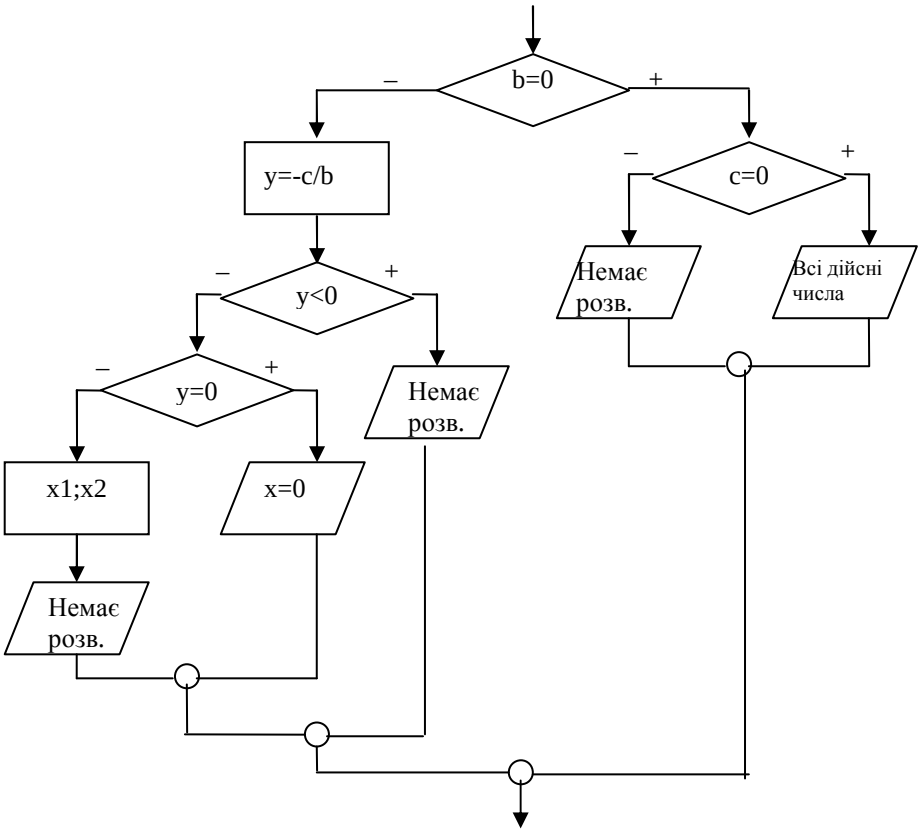

Рис. 5

Розглянемо другий блок (рис. 6).

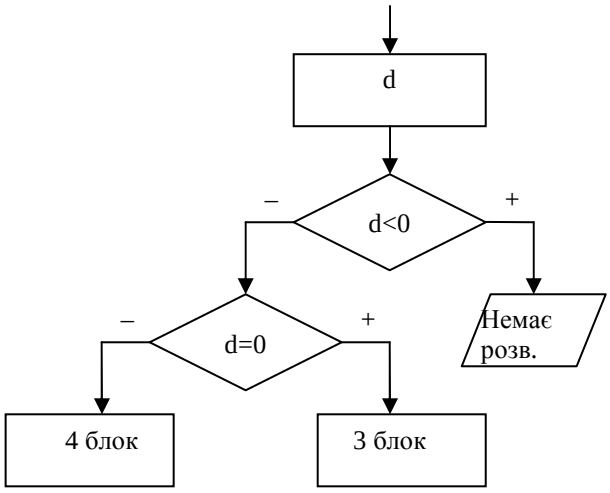

Рис. 6

Як бачимо, другий блок розділився ще на два. Розглянемо тепер кожен з них. 3 блок буде мати вигляд (рис. 7):

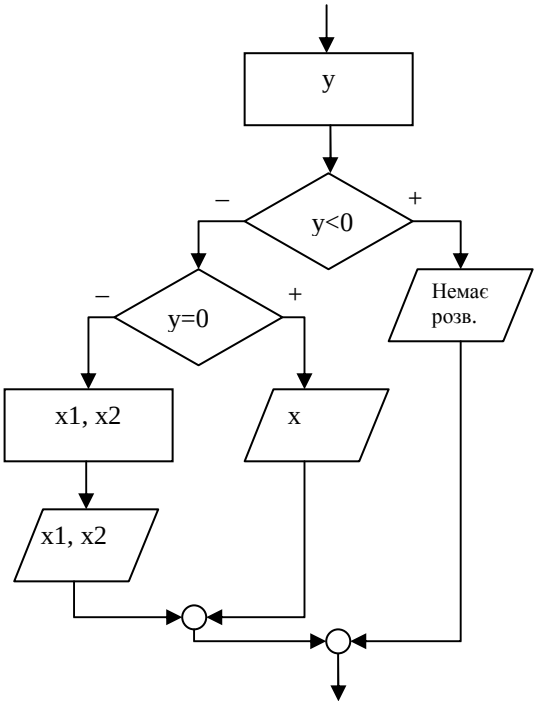

Рис. 7

Четвертий блок прийме вигляд (рис. 8):

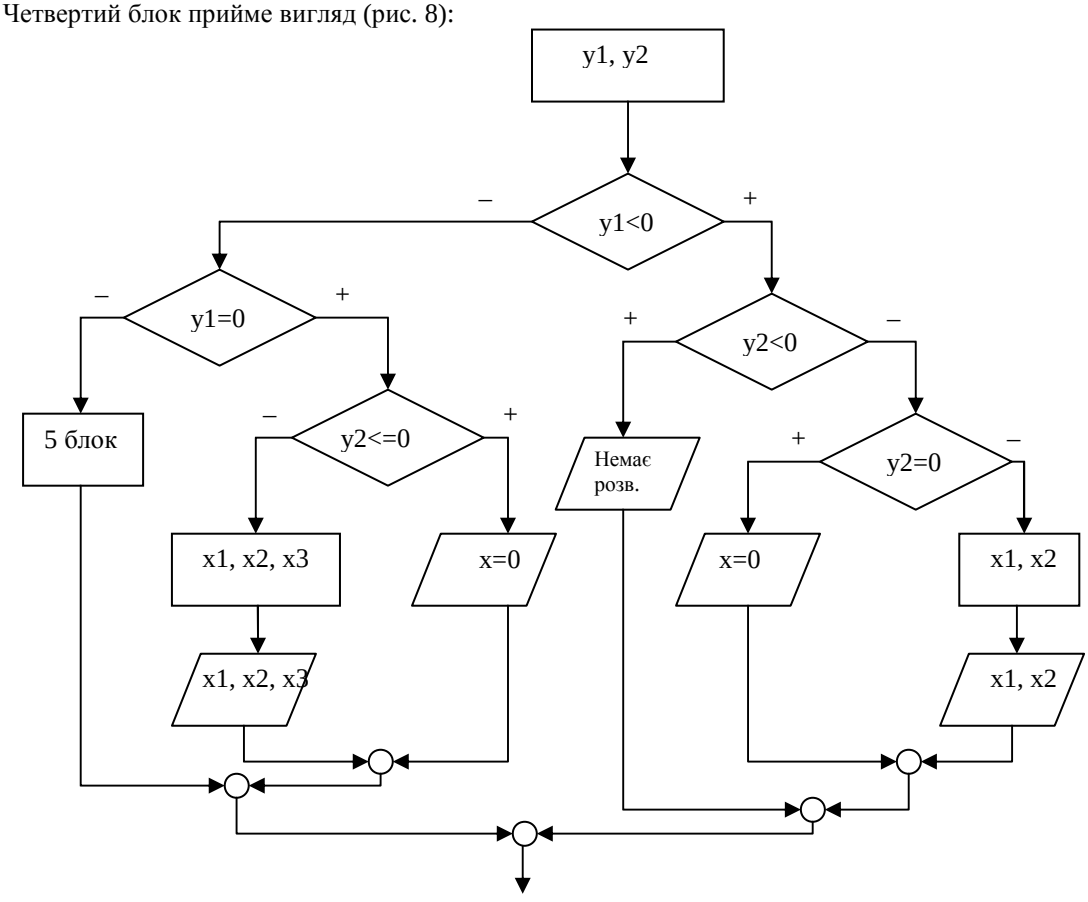

Рис. 8

Як бачимо, у четвертому блоці з’являється ще один п’ятий блок. Розглянемо його (рис. 9):

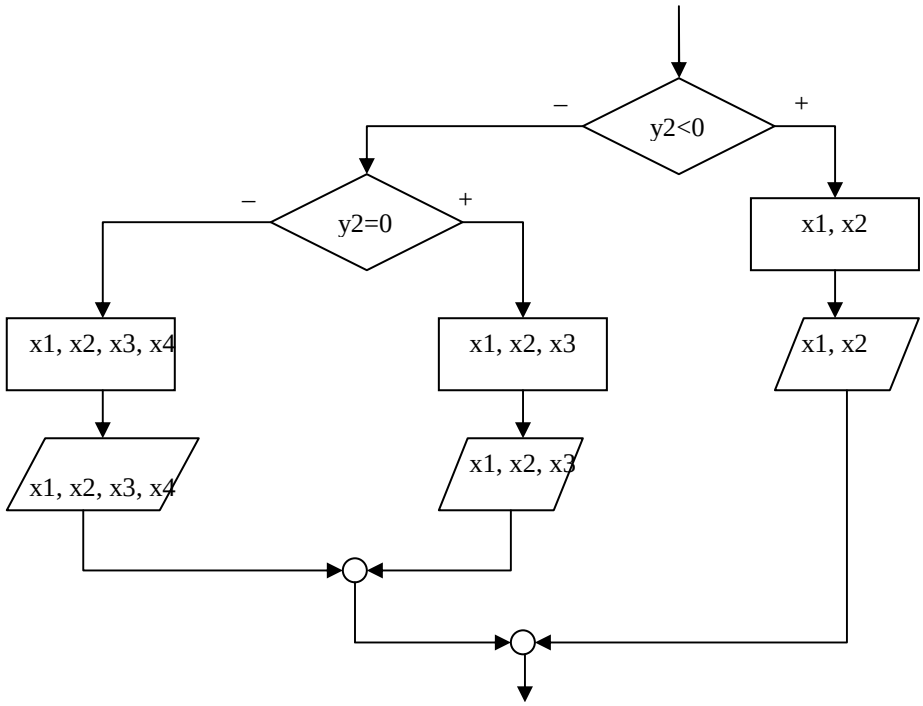

Рис. 9

Зібравши всі компоненти, отримаємо схему алгоритму розв'язування задачі. Реалізуємо розроблену схему алгоритму однією з мов програмування, наприклад Pascal.

```
program bikvadr_rivn;
var a,b,c,d,x,x1,x2,x3,x4,y,y1,y2 : real;
begin
   write('a= '); readln(a);
   write('b= '); readln(b);
   write('c= '); readln(c);
   if a=0 then
{початок першого блоку}
      if b=0 then
            if c=0 then writeln('Rozv e mnowuna vsih dijsnih 4usel')
            else writeln('rozv newmae')
      else begin
              y:=-c/b;
              if y<0 then writeln('rozv newmae')
              else if y=0 then writeln('x=0')
              else begin
                   x1:=sqrt(y); x2:=-sqrt(y);
                   writeln('x1=',x1:6:2,' x2=',x2:6:2);
                  end
         end
   else
{кінець першого блоку}
{початок другого блоку}
   begin
     d:=sqr(b)-4*a*c;
     if d<0 then writeln('rozv nemae')
     else if d=0 then
{початок третього блоку}
                 begin
                    y:=-b/(2*a);
                    if y<0 then writeln('rozv newmae')
              else if y=0 then writeln('x=0')
```

```
            else begin
                 x1:=sqrt(y);  x2:=-sqrt(y);
                 writeln('x1=',x1:6:2,' x2=',x2:6:2);
               end
             end
     else
{кінець третього блоку}
{початок четвертого блоку}
      begin
        y1:=(-b-sqrt(d))/(2*a);  y2:=(-b+sqrt(d))/(2*a);
        if y1<0 then
          if y2<0 then writeln('rozv newmae')
          else if y2=0 then writeln('x=0')
          else begin
                 x1:=sqrt(y2);  x2:=-sqrt(y2);
                 writeln('x1=',x1:6:2,' x2=',x2:6:2);
               end
       else if y1=0 then
              if y2<=0 then writeln('x=0')
              else
              begin
                x1:=0;  x2:=sqrt(y2);  x3:=-sqrt(y2);
                writeln('x1=',x1:6:2,' x2=',x2:6:2,' x3=',x3:6:2);
              end
{початок п’ятого блоку}
        else if y2<0 then
             begin
               x1:=sqrt(y1);  x2:=-sqrt(y1);
               writeln('x1=',x1:6:2,' x2=',x2:6:2);
             end
           else if y2=0 then
                begin
                  x1:=0;  x2:=sqrt(y1);  x3:=-sqrt(y1);
                  writeln('x1=',x1:6:2,' x2=',x2:6:2,' x3=',x3:6:2);
                end
           else
             begin
               x1:=sqrt(y1);  x2:=-sqrt(y1);  x3:=-sqrt(y2);  x4:=sqrt(y2);
               writeln('x1=',x1:6:2,' x2=',x2:6:2,' x3=',x3:6:2,' x4=',x4:6:2);
             end;
        end;
   end;
end.
{кінець п’ятого блоку}
{кінець четвертого блоку}
{кінець другого блоку}
```

Ми навели приклад розв’язання шкільної задачі з математики за допомогою структурного програмування. Проте, незважаючи на подані матеріали, дана проблема залишається в полі зору педагогів-методистів, розробників методичних матеріалів для учнів загальноосвітніх шкіл та студентів вищих навчальних закладів, а тому потребує подальшого вивчення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Державна національна програма "Освіта" України XXI століття. – К. Радуга, 1994. – 50 с.
2. Дал У., Дейкстра Э., Хоор К. Структурное программирование. – М.: Мир, 1973. – 247 с.
3. Дейкстра Э. Дисциплина программирования. – М.: Мир, 1978. – 275 с.
4. Вирт Н. Систематическое программирование. – М.: Мир, 1977. – 183 с.

5. Ляшенко Б. М. Побудова структурованих алгоритмів методом низхідного проектування // Технологія навчання в процесі підготовки майбутнього вчителя. Том II. Частина 1. – Житомир. – 1993. – С. 119-121.
6. Лингер Р., Миллс Х., Уитт Б. Теория и практика структурного программирования: Пер. с англ. – М.: Мир, 1982. – 406 с.

Матеріал надійшов до редакції 19.11. 2009 р.

Присяжнюк Т. А. Использование элементов структурно-логического мышления при решении задач методами нисходящего проектирования.

Определены и проанализированы основные понятия структурного программирования. Предлагаются способы эффективного применения метода структурного программирования в профессиональной деятельности будущего учителя информатики, применены элементы структурно-логического мышления к решению задач средствами нисходящего проектирования. Приведены примеры применения метода нисходящего проектирования к решению задач по школьной информатике.

Prisyazhnyuk T. A. Application of Elements of Structural-Logical Thinking to Decision of Tasks by Facilities of Top-down Design.

The basic concepts of the structured programming are defined and analysed. The ways of effective application of method of the structured programming are offered in professional activity of future teacher of informatics, the elements of structural-logical thought are applied to the decision of tasks by facilities of top-down design. The examples of application of method of top-down design are resulted to the decision of tasks on a school informatics.