УДК 378.147:004.413

Кривонос Олександр Миколайович
старший викладач кафедри прикладної математики та інформатики
Житомирський державний університет імені Івана Франка, м. Житомир, Україна
Alexander.Kryvonos@zu.edu.ua

ВИКОРИСТАННЯ ЗАДАЧНОГО ПІДХОДУ В ПРОЦЕСІ НАВЧАННЯ ПРОГРАМУВАННЯ МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ

Анотація. Стаття присвячена проблемі навчання програмування майбутніх учителів інформатики з позиції компетентністного підходу до навчання. У ній розкрито роль і місце задачного підходу в процесі навчання модуля «Процедурне програмування», що є складовою курсу програмування; розглянуто систематизацію складності рівнів задач, що пропонує Д. М. Толленгерова. Описано рівні складності задач (репродуктивний, частково-пошуковий, дослідницький (творчий)), що використані в побудові методичного забезпечення курсу програмування. А також наведено приклади задач, для конкретних тем, для розв'язання яких студенту необхідно використати навички, які є ключовими для інформаційно-комунікаційно-технологічної компетентності.

Ключові слова: програмування; задача; учитель інформатики; задачний підхід.

1. ВСТУП

Постановка проблеми. Одним із показників, які характеризують сучасне суспільство, є його інформатизація. Володіння навичками роботи з комп'ютером і використання ІКТ у фаховій діяльності ставиться поряд із такими якостями, як уміння читати і писати. На думку В. Ю. Бикова, у діяльності навчальних закладів усіх типів і рівнів акредитації проблема інформатизації повинна посідати першочергове місце [1]. Проблема інформатизації освіти безпосередньо пов'язана з розв'язанням проблеми підготовки вчителів до використання ІКТ у своїй професійній діяльності й формуванням у педагогів поряд з іншими компетентностями інформаційно-комунікаційних. Учитель повинен розуміти, що це дозволить йому інтенсифікувати і покращити свою працю.

У процесі навчання мають формуватися ключові компетентності, інтелектуальні якості — формуватися особистість майбутнього фахівця як єдине ціле. Сучасному вчителю інформатики не достатньо мати навички щодо використання традиційних технологій, а необхідно знати й вивчати освітній сегмент мережі Інтернет, мати навички впровадження ІКТ у навчальний процес, уміти навчати інформатиці за допомогою різних засобів телекомунікації тощо.

Одним з найактуальніших питань у навчанні інформатики є роль і місце програмування в навчальному плані. Під час навчання програмування виникла необхідність використовувати методи підвищеної мотивації навчання студентів комп'ютерних спеціальностей і майбутніх викладачів інформатики для школи.

Під методами навчання програмування розглядаємо впорядковані способи взаємопов'язаної діяльності викладача і студента, спрямовані на досягнення певної мети навчання програмування. Методи навчання відрізняються один від одного критерієм, який взято в основу кожного з них. За способом подання навчального матеріалу виділяють вербальні, наочні та практичні методи навчання. У процесі навчання програмування використовуються вербальні (лекції) і практичні (виконання лабораторних робіт, практикумів, проектів) методи, що дає змогу студентам не лише отримувати нові знання і набувати практичних навичок, але й формувати ключові

© О.М.Кривонос,2014.

компетентності, у тому числі — й інформаційно-комунікаційні. Викладач виступає в ролі інструктора, наголошує на завданнях роботи, скеровує і певною мірою контролює хід її виконання. А діяльність студентів — переважно практична, у якій суттєву роль відіграє самостійний розумовий процес, котрий уможливлює пошук необхідних даних і алгоритмів розв'язування задач.

Аналіз останніх досліджень і публікацій. Завдання викладача у вивченні курсу програмування для майбутніх учителів інформатики — навчити методам побудови ефективних (раціональних) алгоритмів, їх аналізу і реалізації сучасними засобами програмування. Лише знання певної мови програмування, як і наявність будь-якого засобу, нічого не дає майбутньому викладачеві інформатики. Набагато важливішим є вміння використовувати ці знання й навички, самостійно конструювати сам процес розв'язування фахового завдання [2].

Під час вивчення модуля «Процедурне програмування» студенту, окрім ознайомлення з певним теоретичним матеріалом, необхідно виконувати практичні завдання. Йдеться про побудову математичної моделі, розробку алгоритмів, написання коду, проектування структур даних, створення модулів та певних об'єктів із дотриманням певних вимог.

Одним із ключових наукових підходів у контексті досліджень у галузі дидактики вищої педагогічної школи є задачний підхід. Сучасні психологи розкрили провідну роль задач у навчанні — вони є головним чинником підвищення пізнавальної і практичної активності учнів [10].

Мета статі — розкрити місце задачного підходу в процесі навчання програмування майбутніх учителів інформатики.

2. МЕТОДИ ДОСЛІДЖЕННЯ

У процесі формування творчої активності студентів задачний підхід має ґрунтовну теоретичну базу, котра утворилася завдяки інтересу науковців до явищ проблемності: джерел і стимулів мислення, форм наукового пошуку, головних чинників розвитку мотивів і здібностей [5, с. 49].

Аналіз передового педагогічного досвіду дає змогу зробити висновок про те , що головна дидактична мета задачного підходу — організація викладачем процесу засвоєння знань шляхом структурування навчального матеріалу у вигляді послідовності задач, що мають певний логічний зв'язок одна з одною. Студент, слідуючи по цьому ланцюгу від однієї задачі до іншої, включається в активний процес самостійного оволодіння знаннями. Наразі перелік задач досить великий: від найпростіших до дослідницьких підвищеної складності. Д. М. Толленгерова пропонує таку систематизацію складності рівнів задач з програмування:

- задачі, які потребують лише відтворення відомого базового алгоритму;
- задачі, які потребують внесення незначних змін до базового алгоритму і виконання для цього нескладних розумових операцій;
- задачі, які потребують внесення вагомих змін до базового алгоритму і виконання для цього більш складних розумових операцій, що передбачають аналіз ситуації;
- задачі, які потребують логічно обґрунтованого об'єднання різних базових алгоритмів, синтезу відомих алгоритмів і складних розумових операцій;
- задачі, у формулюванні яких відтворено реальні ситуації, що потребують вирішення (дослідницькі задачі підвищеної складності, котрі можуть виконуватися у формі проекту) [13].

Останній тип задач будемо розглядати як компетентнісно-орієнтовані завдання.

У програмуванні, як і в інших дисциплінах природничого циклу, уміння й навички розв'язування задач формуються безпосередньо в процесі виконання цього виду діяльності із застосуванням теоретичних основ навчального предмета, у поєднанні з теоретичними основами суміжних дисциплін (математика, геометрія, статистика тощо). Н. Вірт зазначає, що «в процесі розв'язування будь-якої задачі як за допомогою комп'ютера, так і без нього потрібно обрати деяку абстракцію дійсності, тобто визначити перелік даних, які описують реальну ситуацію. Цей вибір залежить від задачі, котру необхідно розв'язати. Потім слід обрати спосіб подання цих даних» [1].

Під методом розв'язання задачі в програмуванні розуміють сукупність прийомів розумової діяльності або логічних операцій, за допомогою яких розв'язується широкий клас задач, а під способом розуміють сукупність прийомів розумової діяльності або логічних дій для розв'язання конкретної задачі (за аналогією з математикою).

Вітчизняний психолог В. О. Моляко відокремлює ще один термін — «стратегія». Він не розглядає терміни «метод», «спосіб» та «стратегія» як синоніми, оскільки метод і спосіб, на відміну від стратегії, абстраговані від особистості. Термін «стратегія» може бути застосований до «опису всього процесу розв'язування, у якому реалізується динамічна тенденція розумової діяльності особистості стосовно конкретної задачі» [9, с. 17]. Він виділяє такі компоненти структури стратегії: вивчення умови задачі; пошук шляху розв'язання; втілення гіпотези розв'язання.

Власне, на базі основних методів і прийомів розв'язування, а також впевненості у своїх діях формується стратегія розв'язування задачі. У програмуванні про стратегію можна говорити під час розв'язування задач на зразок задач Баше де Мезіріака або ігрових задач, у яких необхідно обрати виграшну стратегію.

Звернемо увагу на те, що будь-яка задача або вправа, котра пропонується студенту в межах компетентнісного підходу, може бути розглянута і як діагностуюча і, як формуюча. Тобто, коли пропонується розв'язати ту чи іншу задачу, викладач може виявити в студента певні «прогалини в знаннях» і сформовані навички, а також оцінити рівень їхнього розвитку.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для перевірки ефективності використання задачного підходу в процесі формування і розвитку інформаційно-комунікаційних компетентностей майбутніх учителів інформатики під час навчання курсу програмування було розроблено програму й методику наукового пошуку. Педагогічний експеримент проводився на базі фізико-математичного факультету Житомирського державного університету імені Івана Франка. Студенти, які брали участь в експерименті, вивчали курс програмування згідно з навчальною програмою, яка відповідала вимогам стандартів вищої освіти.

У ході експерименту студентам пропонувались завдання, що вимагали знань і досвіду роботи в галузі інформаційних технологій. На лабораторному занятті з теми «Лінійні програми» студентам пропонується скласти алгоритм і написати програми для розв'язання поданої далі задачі.

У кафе встановлено Wi-Hi точку доступу на основі роутера (вказується конкретна модель). Яка буде швидкість доступу до мережі Інтернет, якщо N відвідувачів кафе користуються ноутбуками.

Для уточнення умови задачі студентам необхідно в мережі Інтернет знайти технічні параметри роутера, а саме — стандарт, який підтримує цей пристрій. Враховуючи стандарт, визначається максимальна швидкість обміну даними. Далі

необхідно це максимальне значення поділити на кількість користувачів (усі користувачі мають рівні права доступу).

На лабораторному занятті з теми «Циклічні алгоритми» студентам пропонується для розв'язування така задача.

Для введення пінкоду банківської картки використовують цифри від 0 до 9. Обчислити кількість можливих варіантів, якщо пінкод — це N-значне число, яке складається з K різних цифр 0<K<N.

У студентів під час розв'язання зазначеного завдання проводиться аналогія зі ступенем системи захисту даних, а саме — персональних даних користувача інформаційними технологіями.

Під час вивчення зазначеної теми студенти також виконують завдання, яке передбачає наявність певних знань із топології комп'ютерних мереж.

Тубу витої пари, довжина якої N метрів, необхідно розділити на пачкорди різної довжини у такий спосіб, щоб із трьох будь-яких пачкордів неможливо було утворити трикутник (пачкорд розглядається як відрізок певної довжини).

Математична модель цієї задачі зводиться до ряду Фібоначі. Кожен наступний член послідовності дорівнює сумі двох попередніх, і тому для будь-якої трійки елементів цієї послідовності не виконується умова трикутника. З іншого боку довжина пачкорду, за стандартами повинна бути кратна 0,5 м. Тому перші елементи ряду будуть 0,5 і 0,5, а довжина останнього пачкорду дорівнює сумі останнього члена ряду і довжини залишку витої пари. У програмі необхідно зробити перевірку на кратність 0,5 для числа *N*. Код програми на відповідних мовах програмування матиме вигляд:

Pascal

```
Var
 a,b,c,n,s: real;
Begin
 Readln(n);
 s:=n/0.5;
a:=int(s)*0.5;
 if (n>=0.5) and (a=n)
then
   begin
    a:=0.5; b:=0.5;
    s:=a+b;
    while s<n do
     begin
       write(a:0:1,' ');
       c:=a+b;
       a:=b; b:=c;
       s:=s+b;
     end;
    S:=s-b-a;
    b:=n-s;
    writeln(b:0:1);
   end
   else writeln('Error')
End.
```

Python

```
n = input()
s = n / 0.5
a = int(s) * 0.5
if n >= 0.5 and a == n:
    a = b = 0.5
    s = a + b
    while s < n:
        print a, '  ',
        c = a + b
        a = b
        b = c
        s += b
    s -= a + b
    b = n - s
    print b
else:
    print 'Error'
```

C++

```
#include <iostream.h>
int main()
{
  float a,b,c,n,s;
  cin>>n;
  s=n/0.5; a=int(s)*0.5;
  if (n>=0.5 && a==n)
   {
     a=0.5; b=0.5;
     s=a+b;
     while (s<n)
       {
         cout<<a<<" «;
         c=a+b;
         a=b; b=c;
         s=s+b;
       }
     s=s-b-a;
     b=n-s;
     cout<<b;
   }
  else cout<<"Error";
  return 0;
}
```

На лабораторному занятті з теми «Підпрограми» студентам пропонується для розв'язування така задача.

Скласти програму, яка перевіряє гіпотезу Гольдбаха.

В умові задачі навмисно відсутнє формулювання гіпотези. Студенти, використовуючи пошукові системи, повинні знайти формулювання гіпотези Гольдбаха на сайті, який заслуговує довіри, наприклад, на сторінці Вікіпедії є формулювання:

«У математиці гіпотезою Гольдбаха називається наступне твердження:

Довільне парне число не менше чотирьох можна подати у вигляді суми двох простих чисел.

4=2+2;

6=3+3;

8=3+5;

10=3+7=5+5;

12=5+7;

14=3+11=7+7;

16=3+13=5+11;

і так далі.

Довільне непарне число не менше семи можна записати у вигляді суми трьох простих чисел.

Наприклад:

7=2+2+3;

9=3+3+3;

11=2+2+7=3+3+5

і так далі.»

На лабораторному занятті з теми «Відображення даних у графіці» студентам пропонується для розв'язування наступна задача.

Побудувати криву відповідно до варіанту а) локон Аньєзи; б) кардіоїда тощо.

В умові задачі знову навмисно відсутній математичний опис формули кривої. Студенти, використовуючи пошукові системи, повинні знайти параметричний опис формули кривої відповідно до варіанту на сайті, який заслуговує довіри, наприклад на сторінці geometrie.ru знаходимо такі параметричні формули:

Верзьєра (верзієра) Аньези (або ло́кон Аньєзи) — плоска крива, геометричне місце точок M, для котрих виконується співвідношення $\frac{BM}{BC}=\frac{OA}{OB}$, у якому OA — діаметр кола, BC — півхорда цього кола, перпендикулярна до OA.

Рівняння

O = (0,0), A = (0,a)

У прямокутній системі координат: $y=\frac{a^3}{a^2+x^2}$

Параметричне рівняння: $\begin{cases} x = atg\varphi \\ y = a\cos^2\varphi \end{cases}$, φ — кут між OA і OC

Кардіо́їда (греч. καρδία — серце, грец. ετδος — вид) — плоска лінія, котру накреслює фіксована точка кола, яке котиться по нерухомому колу з таким самим радіусом. Свою назву отримала завдяки схожості зі стилізованим зображенням серця.

Кардіоїда є частинним випадком равлика Паскаля, епіциклоїди та синусоїдальної спіралі.

Рівняння:

У прямокутних координатах:

$\left(x^2+y^2\right)^2-2ax\left(x^2+y^2\right)-a^2y^2=0$

У прямокутних координатах (параметричний запис):

$x=a\left(2\cos t-\cos 2t\right)$

$y=a\left(2\sin t-\sin 2t\right)$

У полярних координатах:

$r=a\left(1-\cos\varphi\right)$.»

Результат побудови студенти повинні порівняти з результатом побудови зазначених плоских кривих в інших програмних середовищах, таких як Gran або DG.

Задачний підхід у процесі вивчення модуля «Процедурне програмування» базується на побудові інформаційної (математичної) моделі задачі, побудові алгоритму її розв'язання і лише потім на реалізації алгоритму певною мовою програмування. Мова й середовище програмування можуть бути замінені на інші, якщо в цьому виникне необхідність.

У процесі розв'язування пропонованих завдань студенти виявляють навички, які є ключовими для інформаційно-комунікаційно-технологічної компетентності, відомими під назвою «велика сімка» [1]. Вони показують, як універсальні навички пошуку й перетворення даних з допомогою комп'ютера та інших засобів ІКТ можуть бути інтегровані в систематичний процес, орієнтований на розв'язання практичного завдання. До цього переліку належать: **визначення (ідентифікація даних)** — визначити умову задачі, ідентифікувати необхідні дані; **управління** — виявити всі можливі джерела даних і відібрати серед знайдених ті, котрі найбільше відповідають проблематиці задачі; **пошук даних** — знайти необхідне джерело даних і необхідний матеріал всередині джерела; **інтеграція** — порівняти і зіставити відомості з різних джерел, подавати одержані результати належним чином; **оцінка** — знайдені дані відповідають критеріям відбору і правильно оцінені ресурси, затрачені на цей пошук; **створення** — розв'язувати задачу на основі наявного матеріалу або створювати нові дані; **передавання даних** — передавання повідомлень або відомостей із використання сучасних ІКТ.

Проектуючи і використовуючи систему навчальних завдань, викладач реалізує диференційований підхід, оскільки система завдань містить три рівні складності, що дає змогу не тільки кількісно оцінити навчальний результат, а й студенту самостійно обрати індивідуальну траєкторію навчання.

Ретельно продуманий за змістом і послідовністю широкий спектр різноманітних видів і типів навчальних завдань з програмування різного рівня складності дає змогу інтенсифікувати процес навчання, сприяє його високій результативності у підготовці майбутніх учителів інформатики.

У ході розробки практичної частини лабораторної роботи дотримувалися таких положень:

- складність завдань збільшується поступово, оскільки ступінь формування інформаційно-комунікаційної компетентності студентів під час виконання завдань змінюється;
- самостійність і активність студентів досягається шляхом використання ІКТ під час пошуку необхідного теоретичного матеріалу;
- формування інформаційно-комунікаційних компетентностей майбутніх учителів інформатики забезпечується наявністю в завданнях мотивації до творчого виконання і пошуку, опанування додаткових даних;
- наявність вправ, які відкривають простір для поглиблення й розширення знань, умінь та навичок, професійних здібностей і творчого мислення (компетентнісно-орієнтовані завдання).

Методичне забезпечення містить завдання трьох рівнів: репродуктивного, частково-пошукового, дослідницького (творчого). Щоб виконати завдання будь-якого рівня, крім репродуктивного, як правило, необхідно зробити завдання попереднього рівня, адже перехід від попереднього рівня до наступного відбувається шляхом зростання складності матеріалу.

Завдання репродуктивного рівня містять тренувальні вправи, для виконання яких достатньо лекційного матеріалу. Студентам пропонується скласти нескладний

алгоритм на основі теоретичного матеріалу, реалізувати його на одній з мов програмування (Pascal, C або інша мова за вибором студента), провести тестування програми, перевірити одержані результати.

Такого роду завдання не достатньо ефективні для формування інформаційно-комунікаційних компетентностей студентів, але вони дозволяють швидко формувати практичні навички й уміння, виявляють типові помилки роботи в середовищі або під час складання алгоритмів й працювати над їх виправленням. Завдання цього типу розв'язуються за певним алгоритмом або зразком, наприклад*: упорядкувати масив цілих чисел за зростанням, використовуючи вказаний алгоритм*. Це завдання необхідно виконувати всім студентам (із різним рівнем сформованості інформаційно-комунікаційної компетентності).

Частково-пошукові навчальні завдання передбачають розв'язування вправ із використанням заданого алгоритму, який необхідно певною мірою модифікувати; взаємопов'язаних задач, у яких поступово ускладнюється умова. Складність завдань від модуля до модуля поступово нарощується. Наприклад: *впорядкувати масив дійсних чисел за зростанням кількості розрядів у дробовій частині, використовуючи вказаний алгоритм.* Виконуючи їх, студенти привчаються до самостійної роботи й пошуку додаткового матеріалу, зокрема в мережі Інтернет на спеціалізованих сайтах. Така діяльність спрямовує їх на пошук власного способу виконання навчального завдання.

Дослідницькі завдання відповідають достатньому рівню сформованості інформаційно-комунікаційної компетентності майбутніх учителів інформатики. До такого роду завдань відібрані нестандартні вправи, алгоритм розв'язання яких неможливо описати без детального аналізу умови й використання додаткових джерел даних. Наприклад: *з усіх цифр цілого числа утворити найменше число, кількість розрядів не перевищує 100.*

4. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Використання для навчання програмування задачного підходу, елементів екстремального, парного програмування дозволяє зосередити увагу студентів на розв'язуванні задач і «знімає бар'єр» для вивчення нових мов програмування, що є важливим для майбутніх учителів інформатики.

На жаль, сьогодні існує великий якісний розрив між програмними продуктами, які студенти створюють самостійно, і тими, які вони використовують під час навчання чи роботи за комп'ютером. Проте власноруч створена програма з використанням об'єктно-орієнтованих технологій, в основі якої закладено унікальний алгоритм, може з гідністю конкурувати на ринку програмних засобів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Биков В. Ю. Сучасні завдання інформатизації освіти / В. Ю. Биков // Інформаційні технології і засоби навчання: електронне наукове фахове видання [Електронний ресурс] / Ін-т інформ. технологій і засобів навчання АПН України, Ун-т менеджменту освіти АПН України; гол. ред.: В. Ю. Биков. – 2010. – № 1(15). — Режим доступу : URL :http://www.ime.edu-ua.net/em15/emg.html. — Назва з екрану.
2. Борисова Е. А. Из опыта обучения программирования на занятиях по информатике в экономическом вузе / Е. А. Борисова // Проблемы и перспективы развития образования: материалы междунар. заоч. науч. конф., (Пермь, апрель 2011 г.). —Т. II. — Перм : Меркурий, 2011. — С. 45–47.
3. Бурмакина В. Ф. Большая Семёрка (Б7). Информационно-коммуникационно-технологическая компетентность. Методическое руководство для подготовки к тестированию учителей

[Электронный ресурс] / В. Ф. Бурмакина, М. Зелман, И. Н. Фалина. — М., 2007. — Режим доступу : URL : http://ifap.ru/library/book360.pdf. — Название с экрана.
4. Вирт Н. Программирование на языке Модула-2 / Н. Вирт [пер. с англ.]. — М. : Мир, 1987. — 224 с.
5. Корнієнко С. В. Розв'язування психологічних задач як спосіб становлення майбутнього вчителя : дис. … канд. псих. наук : 19.00.01 / Корнієнко С. В. — К., 2008. — 172 с.
6. Кривонос О. М. Методичне забезпечення творчої навчальної діяльності студентів у процесі навчання основ інформатики / О. М. Кривонос // Вісник Житомирського державного університету імені Івана Франка. — 2005. — Вип. 20. — С. 87–91.
7. Кривонос О. М. Організація самостійної роботи бакалаврів математики з курсу «Програмування» [Електронний ресурс] / О. М. Кривонос // Інформаційні технології і засоби навчання. — 2009. — № 2 (10). — Режим доступу до журн. : http://journal.iitta.gov.ua.
8. Кривонос О. М. Особливості викладання програмування у вищому навчальному закладі з врахування вимог сучасності / О. М. Кривонос // Вісник Житомирського державного університету імені Івана Франка. — 2011. — Вип. 57. — С. 131–134.
9. Моляко В. А. Психология конструкторской деятельности / В. А. Моляко. — М. : Машиностроение, 1983. — 136 с.
10. Сидоренко Е. В. Методы математической обработки в психологи / Е. В. Сидоренко. — СПб. : ООО «Речь», 2003. — 350 с.
11. Спірін О. М. Початки алгоритмізації та процедурного програмування : метод. посіб. для студ. вищих пед. навч. закл-ів фізико-математичних спец-тей / О. М. Спірін, О. М. Кривонос. — Житомир: ЖДПУ, 2002. — 93 с.
12. Спірін О. М. Теоретичні та методичні засади професійної підготовки майбутніх учителів інформатики за кредитно-модульною системою [монографія] / Олег Михайлович Спірін; [наук. ред. М. І. Жалдак] / Житомирський держ. ун-т ім. Івана Франка. — Житомир : Видавництво ЖДУ ім. І. Франка, 2007. — 300 с.
13. Толлингерова Д. Психология проектирования умственного развития детей / Д. Толлингерова. — М.: Прага, 1994. —48 с.

Матеріал надійшов до редакції 14.02. 2014 р.

ИСПОЛЬЗОВАНИЕ ЗАДАЧНОГО ПОДХОДА В ПРОЦЕССЕ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЯ БУДУЩИХ УЧИТЕЛЕЙ ИНФОРМАТИКИ

Кривонос Александр Николаевич
старший преподаватель кафедры прикладной математики и информатики
Житомирский государственный университет имени Ивана Франко, г. Житомир, Украина
Alexander.Kryvonos@zu.edu.ua

Аннотация. Статья посвящена проблеме обучения программирования будущих учителей информатики с позиции компетентностного подхода к обучению. В ней раскрыто роль и место задачного подхода в процессе изучения модуля «Процедурное программирование», который является составляющим курса программирования; рассмотрено систематизацию сложности уровней задач, которую предлагает Д. М. Толленгерова. Описано уровни сложности задач (репродуктивный, частично-исследовательский, исследовательский (творческий)), которые использованы в построении методического обеспечения курса программирования. Также приведены примеры задач, для конкретных тем, для решения которых студенту необходимо использовать навыки, которые являются ключевыми для информационно-коммуникационно-технологической компетентности.

Ключевые слова: программирование; задача; учитель информатики; задачный подход.

USING OF TASK APPROACH METHOD WHILE TEACHING PROGRAMMING TO THE FUTURE INFORMATICS TEACHERS

Oleksandr M. Kryvonos
Senior Lecturer, Department of Applied Mathematics and Informatics
Zhytomyr Ivan Franko State University, Zhytomyr, Ukraine

Alexander.Kryvonos@zu.edu.ua

Abstract. This article is dedicated to the problem of teaching programming to the future informatics teachers from the standpoint of competence approach in teaching. The article defines the role and the place of task approach in the process of teaching the module on “Procedure programming”, which is the part of the programming course; it scrutinizes the systematization of levels of tasks, which are proposed by D. Toleengerov. The article describes the levels of complexity of tasks (reproductive, partially searching, research (creative)), which are used in the formation of methodological provision for programming course. It also presents the examples of tasks of specific topics to solve which a student should have habits which are crucial for informational communicational technological competence.

Keywords: programming; scientific problem; teacher of informatics; method of scientific problem.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Bykov V. Yu. Modern Tasks of Educational Informatization [online] / V. Yu. Bykov // Informatsiini tekhnolohii i zasoby navchannia: elektronne naukove fakhove vydannia / In-t inform. tekhnolohii i zasobiv navchannia APN Ukrainy, Un-t menedzhmentu osvity APN Ukrainy; hol. red.: V. Yu. Bykov. — 2010. — # 1(15). — Available from : http://www.ime.edu-ua.net/ em15/emg.html. (in Ukrainian)
2. Borisova E. A. From the Experience of Programming Learning on the Lessons of Informatics in the Economic Educational Establishment / E. A. Borisova // Problemy i perspektivy razvitija obrazovanija: materialy mezhdunar. zaoch. nauch. konf., (Perm', aprel' 2011 g.). —T. II. — Perm': Merkurij, 2011. — S. 45–47. (in Russian)
3. Burmakina V. F. Great Seven (G7). Informational-Communicational-Technological Competence'. Metodicheskoe rukovodstvo dlja podgotovki k testirovaniju uchitelej [online] / V. F. Burmakina, M. Zelman, I. N. Falina. — M., 2007. — Available from: http://ifap.ru/library/book360.pdf. (in Russian)
4. Virt N. Programming in the Language of Module 2 / N. Virt [per. s angl.]. — M. : Mir, 1987. — 224 s. (in Russian)
5. Korniienko S. V. Psychological Tasks Solution as the Means of Future Teacher Formation : dys. … kand. psykh. nauk : 19.00.01 / Korniienko S. V. — Kyiv, 2008. — 172 s. (in Ukrainian)
6. Kryvonos O. M. Methodical Software of the Students' Creative Educational Activity in the Process of Informatics Fundamentals Learning / O. M. Kryvonos // Visnyk Zhytomyrskoho derzhavnoho universytetu imeni Ivana Franka. — 2005. — Vyp. 20. — S. 87–91. (in Ukrainian)
7. Kryvonos O. M. Bachelors' Independent Work Organization on Mathematics in the Course "Programming" [online] / O. M. Kryvonos // Informatsiini tekhnolohii i zasoby navchannia. — 2009. — # 2 (10). — Available from: http://journal.iitta.gov.ua. (in Ukrainian)
8. Kryvonos O. M. Peculiarities of Programming Teaching in Higher Educational Establishment Taking into Account Modern Requirements / O. M. Kryvonos // Visnyk Zhytomyrskoho derzhavnoho universytetu imeni Ivana Franka. — 2011. — Vyp. 57. — S. 131–134. (in Ukrainian)
9. Moljako V. A. Psychology of the Constructive Work / V. A. Moljako. — M. : Mashinostroenie, 1983. — 136 s. (in Russian)
10. Sidorenko E. V. Methods of Mathematical Processing / E. V. Sidorenko. — SPb. : OOO «Rech'», 2003. — 350 s. (in Russian)
11. Spirin O. M. Beginning of Algorithmization and Procedural Programming: metod. posib. dlia stud. vyshchykh ped. navch. zakl-iv fizyko-matematychnykh spets-tei / O. M. Spirin, O. M. Kryvonos. — Zhytomyr : ZhDPU, 2002. — 93 s. (in Ukrainian)
12. Spirin O. M. Theoretical and Methodical Bases of Future Informatics Teachers' Professional Preparation on the Credit-Module System [monohrafiia] / Oleh Mykhailovych Spirin; [nauk. red. M. I. Zhaldak] // Zhytomyrskyi derzh. un-t im. Ivana Franka. — Zhytomyr : Vydavnytstvo ZhDU im. I. Franka, 2007. — 300 c. (in Ukrainian)
13. Tollingerova D. Psychology of Children's Mental Development / D. Tollingerova. — M. : Praga, 1994. — 48 s. (in Russian)