

5. Карпова Л.Г. Формування професійної компетентності вчителя загальноосвітньої школи. Автореф. ...канд. пед. наук: Харківський державний педагогічний університет ім. Г.С. Сковороди. – Харків, 2004. – 20 с.
6. Куликова Л.Н. Проблемы саморазвития личности. – Хабаровск, 1997.
7. Маркова А.К. Психологические проблемы повышения квалификации // Педагогика. – 1992. – № 9-10.
8. Монтень М. Опыты. Избранные главы: Пер. с фр. /Сост. Г.Косикова. – М., 1991.
9. Сухомлинский В.А. Рождение гражданина. – М., 1979.
10. Турчанинова Ю.И. Теоретические проблемы содержания профессиональной подготовки учителя // Педагогическое образование: содержание, структура, организационные формы: Сб. научн. тр. / Под ред. В.Б. Новичкова. – М., 1987.
11. Ушинский К.Д. Избранные педагогические сочинения: В 2-х т. – М., 1974.
12. Харламова Г.С. Роль философии в развитии содержания педагогического образования взрослых. Личностно ориентированное обучение и его педагогический смысл // Современные ориентиры в образовании педагогов. Материалы научн.-практ. конф. – СПб. – Иркутск: ИОВ РАО, 1998.
13. Хохлов С.И., Лосавио О.В. Педагогическая техника и психорегуляция в учебном процессе // Советская педагогика. – 1989. – № 9.
14. Черноусова О.Р. Индивидуальный подход к совершенствованию педагогического мастерства учителя. Дисс. ...канд. пед. наук. – Минск, 1993.

УДК 378.147:681.3

Абдураманов З.Ш., Сейдаметова З.С.

SP-КОМПЕТЕНТНОСТІ: НОВА ПАРАДИГМА ПІДГОТОВКИ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ

Визначені поняття компетентності, системного програмування та ін. Описані когнітивні і практичні компетентності системного програмування, які є важливими складовими в підготовці майбутніх інженерів-програмістів. Визначена група дисциплін, які формують SP-компетентності.

We define the notations of competence, system programming and etc. We describe cognitive and practical competences of the system programming, which are an important constituent in engineers-programmers education. We identify the group of courses forming the SP-competences.

Одним із наріжних каменів нової парадигми освіти, що складається, є варіативна розвиваюча освіта. На відміну від традиційної освітньої моделі, орієнтованої на репродукцію, нова парадигма базується на рішенні проблемних ситуацій, учить індивідуума вільно мислити в світі, що змінюється. Необхідність змін в освіті в сучасному світі, формування нових підходів до навчання відзначають не тільки в пострадянських країнах, на ці питання звертають увагу і в європейських країнах. Багато освітніх проектів Європейського Союзу розглядають нові підходи і методи навчання у вищій освіті, що базуються на компетентнісному підході.

У освітньо-кваліфікаційній характеристиці випускника по напрямку 0802 “Прикладна математика” Галузевого стандарту вищої освіти України позначені цілі освітньої і професійної підготовки студентів спеціальності “Інформатика” освітнього рівня бакалавр; у цьому ж документі є вимоги, що пред’являються до компетентності студентів цієї спеціальності [1].

У роботах А.М. Терехова і Д.Ю.Буличева, В.М. Пільщикова, А.П.Побегайло та ін. розглянуті питання, пов’язані з підготовкою студентів комп’ютерних спеціальностей в області системного програмування [2; 3].

У роботі [8] містяться рекомендації до підготовки фахівців в області комп'ютеринга і складанню типових учбових планів з урахуванням різних моделей навчання. У документах [5; 7] визначені вхідні і вихідні компетентності бакалаврів за базовими профілями підготовки комп'ютеринга з урахуванням сучасних вимог.

Останні дослідження і публікації свідчать, що багатьма ученими активно обговорюються питання модернізації освіти на компетентнісній основі (В.А.Болотов, Т.М. Ковальова, Е.Я. Коган, Н.В. Кузьміна, В.В. Лаптев, О.Е. Лебедев і ін.).

Ціль даної статті полягає у визначенні компетентносній системного програмування майбутніх інженерів-програмістів. Для позначення цих компетентностей зручно використовувати термін *SP-компетентності* (SP – Systems Programming).

Термін “компетентність” визначається в різних джерелах по-різному. Приведемо декілька визначень, які узяті нами за основу.

Компетентність – сукупність знань і умінь, необхідних для ефективної професійної діяльності: уміння аналізувати, передбачати наслідки професійної діяльності, використовувати інформацію [9: 22].

Компетентність – це знання і уміння, необхідні людині в певній області діяльності [10: 18].

Компетентність – не тільки і не стільки наявність і значний об'єм знань і досвіду, скільки уміння їх актуалізувати в потрібний час і використовувати в процесі реалізації своїх службових функцій [11: 346].

Для визначення SP-компетентностей інженера-програміста, визначимо поняття “*Системне програмування*”. *Системне програмування* – діяльність зі створення системного програмного забезпечення. Головною відмінною характеристикою системного програмування в порівнянні з прикладним програмуванням є те, що системне програмування вимагає розуміння і знання характеристик апаратного забезпечення. Важливо відзначити, що програміст, знаючи про специфічні технічні характеристики системи, зможе написати ефективнішу програму, використовувати алгоритм, що дозволяє результативно використовувати комп'ютер.

У програмуванні на апаратному рівні використовуються так звані *низькорівневі мови програмування*. Низькорівневі мови використовуються зазвичай при написанні системних застосувань в ресурсозалежних системах, що дозволяє уникати динамічного переповнювання, а також надає можливість прямого контролю за доступом до пам'яті і потоками, що управляють. У епоху зародження програмування системні програмісти писали програми виключно на низькорівневій мові – асемблері.

Існуюче поняття “*Мова системного програмування*” використовується для розділення понять програмування системного програмного забезпечення і програмування прикладного програмного забезпечення. На відміну від прикладних мов, мови системного програмування зазвичай пропонують прямий доступ до фізичних (технічних) засобів. Відмінності між мовами системного програмування і мовами прикладних застосувань зараз частково стерлися у зв'язку з широким розповсюдженням сучасних мов, наприклад, C/C++, C#, Java та ін.

Джонатан Ярден (Jonathan Yarden, USA – старший адміністратор Unix системи, керівник служби безпеки мережі, старший архітектор програмного забезпечення регіонального Інтернет-провайдера) опублікував в розсилці Internet Security сайту TechRepublic.com 12 травня 2003 року невелику нотатку про те, як повинен мислити програміст, щоб уникнути помилок при написанні їм програм. Зокрема, в ній мовиться: “Часто зустрічаються програмісти, не знайомі з мовою асемблера. Вони витрачають тижні і місяці на відладку програми, намагаючись знайти помилку, із-за якої переповнюється буфер і програма висне. Отже, ви не станете хорошим програмістом без розуміння того, як комп'ютер обробляє коди. Іншими словами, ви повинні думати як комп'ютер” [12].

Таким чином, здатність мислити як комп'ютер допоможе інженерові-програмістові при розробці програмних продуктів і при створенні програмного коду. Здатність правильно

розуміти, як код додатку переводиться в машинний код, допоможе уникнути різних непорозумінь в програмі.

Враховуючи вимоги до компетенції, підготовки фахівців, що склалися в світовій практиці, з комп'ютерних наук, ми вважаємо дуже важливим формування компетентностей системного програмування (SP-компетентності). Для підготовки кваліфікованого фахівця, конкурентноздатного на IT-ринку, ці компетентності необхідно розглядати як найважливішу складову процесу навчання інженерів-програмістів. На нашу думку, формування SP-компетентностей здійснюється групою дисциплін, серед яких “Архітектура комп'ютера” (3 семестр), “Спеціалізовані мови програмування” (4 семестр), “Операційні системи” (5 семестр), “Системне програмування” (6, 7 семестри). Ці дисципліни спільно формують ту групу SP-компетентностей, яку Дж. Ярден лапідарно позначив словами “думати як комп'ютер”.

Дисципліни “Архітектура комп'ютера” і “Спеціалізовані мови програмування” допомагають опанувати теоретичними знаннями про структуру і організацію пам'яті комп'ютера, файлової системи, принципах їх функціонування, про основи роботи процесорів типа Intel x86, щеплять навички низькорівневого програмування в операційних системах DOS, Windows, Unix та ін.

Дисципліни “Операційні системи” і “Системне програмування” знайомлять з теоретичними основами функціонування, фундаментальними принципами проектування і аналізу сучасних операційних систем; дозволяють практично застосовувати отримані знання, а також програмувати пристрої на рівні портів введення/виведення; дають ясне уявлення про основні концепції, структури і механізми різноманітного системного програмного забезпечення (наприклад, операційних системах), найважливіші угоди і конструктивні рішення, використовувані професійними програмістами при розробці системних програм.

SP-компетентності професійні, важливі в реальній виробничій діяльності, але вони можуть виявлятися і в повсякденному житті, в освітньому процесі [13; 14].

Під SP-компетентностями ми розуміємо наступні дві групи – когнітивні і практичні.

1. Когнітивні компетентності:

- Знання і розуміння цифрової логіки і цифрових систем; представлення даних в пам'яті комп'ютера;
- Знання і розуміння організації комп'ютера на рівні асемблера і пам'яті; взаємодії пристроїв;
- Знання і розуміння функціональної організації пристроїв комп'ютера; багатопроцесорної і альтернативної архітектури; сучасної архітектури; напрямів розвитку архітектури процесорів;
- Знання і розуміння архітектури процесорів; конструкцій і методів програмування на мові асемблера; директив, операторів і команд мови Асемблер; основ роботи процесора в захищеному режимі; призначення розширених функцій процесора для реалізації сучасного програмного забезпечення;
- Знання і розуміння операційних систем; обробки процесів, організації багатопотокової обробки, симетричною багатопроцесорності і мікроядер; основних об'єктів операційних систем і набору функцій підтримки їх функціонування; основних тенденцій розвитку комп'ютерних систем; підходів до планування процесів; механізмів захисту від факторів, що представляють загрозу безпеці комп'ютерів;
- Знання і розуміння подієво керованого програмування; основних елементарних функцій роботи з файлами; властивостей об'єкту “файл”, призначення типів змінних при виклику цих функцій; приведення типів змінних; незалежності адресного простору процесів; критичних перетинів як одного із способів синхронізації процесів; відмінностей в клоні Windows з погляду властивостей і атрибутів процесів;

- Знання і розуміння функцій API для створення, знищення і зміни стану процесу; спадкоємство властивостей процесу; принципів основних алгоритмів синхронізації їх достоїнств і обмежень; функцій API зміни стану потоку в процесі, проблем взаємоблокування при реалізації вимог до ресурсів процесу; функцій API для підтримки синхронізації об'єктів операційної системи;
- Моделювання алгоритмів синхронізації на рівні початкового коду; взаємодії системних і призначених для користувача об'єктів операційної системи;
- Моделювання і проектування паралельних систем (диспетчеризація і перемикання контекстів, переривання, паралельного виконання); черг (черги з пріоритетами і без пріоритетів).

2. Практичні компетентності:

- Проектування і реалізація програм на мові Асемблер в реальному і захищеному режимах роботи процесора; застосування різних типів компіляторів; аналіз і відладка програмних кодів на мові Асемблер; дизасемблювання програм, написаних на машинному коді;
- Застосування і використання функцій операційних систем при написанні програм; різних методів роботи з пам'яттю, з дисками і файлами на рівні операційної системи;
- Безпечна інсталяція пакетів з контролем заміни системних бібліотек;
- Застосування і використання властивостей і методів обміну інформацією між додатками на локальному і видалених комп'ютерах;
- Оцінка можливостей операційної системи для реалізації практичних завдань різних типів; перспектив створення багатопотокових застосувань;
- Створення об'єктів синхронізації, їх реалізації в мовах високого рівня, відладки;
- Запуск потоку в процесі створення і аналізу тексту багатопотокових застосувань;
- Контроль правильності виконання програм синхронізації;
- Налаштування операційних систем для вирішення нестандартних завдань за допомогою обґрунтованого підбору існуючого програмного забезпечення і розробки власного; визначення ризиків в питаннях безпеки експлуатації комп'ютерних систем і обладнання;
- Використання спеціального інструментарію при розробці програмних продуктів і повне розуміння процесів, що відбуваються, при рішенні практичних задач.

На сайті освітнього суспільства “Open Source Software” приведені слова великого гурту Дональда Кнута: “Той, хто серйозно цікавиться комп'ютерами, повинен знати про те, чим дійсно є використовуване ними обладнання. Інакше програми, які вони напишуть, будуть дуже чудернацькими” [15]. Ці слова підтверджують необхідність обліку SP-компетентностей при підготовці інженерів-програмістів.

Таким чином, зміни, що сталися в останнє десятиліття в області комп'ютеринга, вимагають модифікації освітніх підходів до підготовки студентів цього напрямку. Сучасний інженер-програміст повинен володіти безліччю спеціальних компетентностей. Одною із складових цього ряду є сформульовані нами SP-компетентності.

Надалі планується розгляд методик формування SP-компетентностей, аналіз різних деменцій SP-компетентності.

ЛІТЕРАТУРА:

1. Освітньо-кваліфікаційна характеристика. Бакалавр. Напрямок підготовки 0802 Прикладна математика // Галузевий стандарт вищої освіти. – К., 2002.
2. Системное программирование: Сб. статей/Под ред. А.Н.Терехова и Д.Ю.Булычева. – СПб., 2004. – 412 с.
3. Пильщиков В.Н. Программирование на языке ассемблера IBM PC. – М.: Диалог-МИФИ, 2001. – 288 с.

4. Побегайло А.П. Системное программирование в Windows. – СПб.: БХВ-Петербург, 2006. – 1056 с.
5. Computing Curricula 2001. Computer Science. – Final Report (December 15, 2001), The Joint Task Force on Computing Curricula, IEEE CS, ACM – 236 p.
6. Software Engineering 2004, Curriculum and Guidelines for undergraduate Degree Programs in Software Engineering. – IEEE CS Press, ACM Press, 2004. – 129 p.
7. Computer Engineering 2004, Curriculum and Guidelines for undergraduate Degree Programs in Computer Engineering. – IEEE CS Press, ACM Press, 2004. – 162 p.
8. Рекомендации по преподаванию информатики в университетах: Пер. с англ. – СПб., 2002. – 372 с.
9. Гушлевська І. Поняття компетентності у вітчизняній та зарубіжній педагогіці // Шлях освіти, №3(33), 2004. – С. 22–24.
10. Корнілова А. Ключові кваліфікації – компетентності особистості у вищих навчальних закладах Німеччини // Шлях освіти. – №3(37). – 2005. – С. 18–22.
11. Журавлев П.В., Карташов С.А., Маусов Н.К., Одегов Ю.Г. Персонал: Словарь понятий и определений. – М.: Экзамен, 1999. – 512 с.
12. Jonathan Yarden. Think like a computer – <http://techrepublic.com>.
13. Сейдаметова З.С. Шляхи адаптації до професії студентів першого курсу комп'ютерних спеціальностей // Комп'ютер у школі та сім'ї. – № 6. – 2002. – С.31-32.
14. Сейдаметова З.С., Темненко В.А. Роль дисципліни “Конкретна математика” в підготовці фахівців з інформатики і прикладної математики: Зб. наук. праць. Педагогічні науки. Випуск 39. – Херсон: Видавництво ХДУ, 2005. – С. 325–330.
15. Softpanorama: Open Source Software Educational Society. – <http://www.softpanorama.org/index.shtml>

УДК 378: 016:78

Балашова О.С.

3 ПИТАНЬ ОПТИМІЗАЦІЇ КОНЦЕРТМЕЙСТЕРСЬКОЇ ПІДГОТОВКИ МАЙБУТНІХ ВЧИТЕЛІВ МУЗИКИ У ВНЗ

У статті розглянуто проблему збагачення змісту концертмейстерської підготовки майбутнього вчителя музики шляхом ознайомлення з методами музично-ритмічного навчання і оволодіння специфікою ритмо-пластичного акомпанементу.

The article considers content enrichment problem of accompanist training of the future teacher of music by force acquaintance of music-rhythmical methods and rhythmoplastical specificity acquirement.

Зміст сучасної музичної освіти вимагає більш ефективного використання концертмейстерських якостей вчителем музики. Спектр їх застосування у шкільній практиці надалі стає ширшим й унеможливує ігнорування. Вони розповсюджуються як на навчальну сферу діяльності вчителя, так і на позакласну. Державним стандартом початкової загальної освіти України, затвердженим Постановою Кабінету Міністрів України (№1717 від 16.11.2000р.) галузь “Мистецтво-Естетична культура” визнано як таку, що розширює “гуманітарний простір в освіті”, “посилює особистісний вимір шляхом радикальної гуманізації”. В основу цієї галузі покладено принцип інтеграції, взаємозв'язку між основними видами мистецтв, а саме: музики, образотворчого мистецтва, хореографії, театрального мистецтва. На сучасному етапі до навчального компоненту початкової школи входить предмет “Музика з елементами пластики”, де єдність та взаємопроникнення музики і руху створюють умови до більш ефективного виховання підростаючого покоління [1].

Отже, виходячи з цього, набуває актуальності проблема спроможності викладачів мистецьких дисциплін, а саме: вчителів музики, використовувати у шкільній практиці