

КОМП'ЮТЕР у ШКОЛІ та СІМ'І

НАУКОВО-МЕТОДИЧНИЙ
ЖУРНАЛ

Виходить 8 разів на рік.
Видається з лютого 1998 року.

Засновники: Інститут педагогіки
АПН України, Інститут інформаційних
технологій і засобів навчання АПН
України, редакція журналу за сприяння
Міністерства освіти і науки України
Свідоцтво про реєстрацію серія КВ
№12217-1101ПР від 17.01.2007.
Передплатний індекс 74248.

Журнал зареєстровано ВАК України
як фаховий у галузі педагогічних наук,
постанова Президії ВАК України
№1-05/7 від 09.06.1999.

Затверджено Вченою радою
Інституту педагогіки АПН України,
протокол №5 від 21.04.2008 р.

Головний редактор
РУДЕНКО В.Д.

Редакційна рада:

ГУРЖИЙ А.М.

ЖИЛЯЄВ І.В.

ЖУК Ю.О.

ЛАПІНСЬКИЙ В.В.

ЛУНЯЧЕК В.Е.

МАШВИЦЬ Ю.І.

МОРЗЕ Н.В.

НАУМЕНКО Г.Г.

ПАСТОВЕНСЬКИЙ О.В.

ПРОКОПЕНКО Н.С.

РАМСЬКИЙ Ю.С.

РЕДЬКО В.Н.

Редакційна колегія:

БИКОВ В.Ю.

БУРДА М.І.

ВАШУЛЕНКО М.С.

ГОЛОВКО М.В.

ЖАЛДАК М.І.

КОРНЄЄВ В.П.

МАДЗИГОН В.М.

СПИВАКОВСЬКИЙ О.В.

ФОКІНА Т.М.

Адреса редакції

04053, Київ-53, а/с 27.

E-mail: csf221@rambler.ru,

csf221@meta.ua.

Тел. (044) 481-37-38.

№3(67) ♦ 2008

ЗМІСТ

ПИТАННЯ ТЕОРІЇ

- Глинський Я.М., Рязьська В.А. Аналіз проектів програм з інформатики для 12-річної школи 3
Лавріщева К.М. Програмна інженерія — спадкоємиця програмування 7

МЕТОДИКА НАВЧАННЯ ІНФОРМАТИКИ

- Горошко Ю.В. Грамбовська Л. Методика вивчення ППЗ GRAN-2D на уроках інформатики та його застосування в планіметрії 14

КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ НАВЧАННЯ

- Парфьонова Н.В., Ларченко Н.В. Електронний підручник Visual Basic 23
Носенко Т.І. Інтерактивні SMART-технології як основний інструмент у викладацькій діяльності 28
Сінько Ю.І. Системи комп'ютерної підтримки практичних знань з математичної логіки «MatLog» 30

КОМП'ЮТЕР В УПРАВЛІНСЬКІЙ ДІЯЛЬНОСТІ

- Сухович Г.А. Прогноз розвитку ЗНЗ на основі комп'ютерних технологій 32

МОБІЛЬНИЙ ІНТЕРНЕТ

- Сіденко О.М. Сучасні комп'ютерні мережі та мобільний зв'язок 38
Погрібний О.В. Робота в Інтернеті за допомогою мобільного телефону 40

ОЛІМПІАДИ З ІНФОРМАТИКИ

- Бондаренко В.В., Галковський Т.О., Коротков А.С., Нейтер Д.Ю., Ягієв Ш.І. Задачі XXI Всеукраїнської олімпіади з інформатики та рекомендації щодо їх розв'язування 45

ІНФОРМАЦІЯ

- XXI Всеукраїнська олімпіада з інформатики 50
Черненко В.О. Європейський форум учителів-новаторів 2008 року 51
Комп'ютерні новини 54
Автори номерів 56

Новий комплект для передплати

З другого півріччя 2008 року читачі можуть отримувати новий передплатний комплект: журнал «Комп'ютер у школі та сім'ї» + газета «Інформатика»,
передплатний індекс 99854.

Вартість комплекту на рік на **20 грн.** дешевше передплати на окремі видання.
Передплата продовжується.

★ ★ ★

Вимоги до статей

Стаття повинна відповідати тематиці журналу і сучасному стану інформатики, бути літературно опрацьованою. Автор статті відповідає за правильність і достовірність викладеного матеріалу, за належність останнього йому особисто.

Обсяг статті—до 10 сторінок машинопису. Структура статті: назва статті, прізвище, ім'я, по батькові автора, текст статті, бібліографія. Автор підписується на звороті останньої сторінки.

Весь текст статті повинен бути набраний шрифтом одного розміру 12 pt через 1,5 інтервали в одну колонку без застосування будь-якого форматування. Не допускається використання вставок та гіперпосилань. Можна виділяти окремі частини тексту курсивом та/або напівжирним шрифтом.

Текст статті друкується на папері стандартного формату з дотриманням таких розмірів полів: верхнього, нижнього і правого—20, лівого—25 мм.

Ілюстративний матеріал виконується чітко і якісно. Посилання на ілюстрації в тексті статті — обов'язкове.

Разом із друкованою статтею надаються дискети або CD, на яких записано статтю. **Екранні копії, схеми, малюнки та фотографії записуються окремими графічними файлами.** Для екранних копій використовуються формати TIF або BMP. Для всіх файлів потрібно вказати назви програм (на звороті останньої сторінки статті), у яких вони створені, наприклад Word 2003, Word XP, CorelDRAW 11.0, Adobe Photoshop 7.0, Adobe Illustrator 8.0 та ін.

До статті додаються відомості про автора: прізвище, ім'я та по батькові повністю, посада, звання, а також домашня адреса, контактні телефони й фото автора (розміром не менше 6x9 см). Стаття, дискети та CD автору не повертаються.

★ ★ ★

Шановні читачі!

Наводимо графік виходу з друку журналу у 2008 році

№3 — 19 травня;	№6 — 30 вересня;
№4 — 30 червня;	№7 — 17 листопада;
№5 — 15 серпня;	№8 — 26 грудня.

Редакція гарантує чітке дотримання наведеного графіка.

АНАЛІЗ ПРОЕКТІВ ПРОГРАМ З ІНФОРМАТИКИ ДЛЯ 12-РІЧНОЇ ШКОЛИ

Глинський Я.М., Ряжська В.А.

Що і як буде вивчатися в курсі шкільної інформатики через декілька років? Це питання хвилює батьків, учителів, методистів і фахівців. У статті зроблена спроба проаналізувати проекти програм з інформатики для 9–12-их класів, які перемогли у конкурсах базових і профільних програм і зараз знаходяться на стадії громадського обговорення та розгляду науково-методичною комісією. Програми опубліковані на сайті Міністерства освіти і науки України. Їх аналіз показує, що методично-організаційні проблеми вивчення інформатики в 12-річній школі не розв'язані повністю.

Нормативно-правова база вивчення інформатики в загальноосвітніх школах наповнена документами, які суперечать один одному. Розглянемо приклади. У концепції загальної середньої освіти (12-річна школа) [1] стверджується: «До складу загальноосвітніх предметів вводиться інформатика. Комп'ютерно-орієнтовані засоби навчання мають застосовуватися на всіх ступенях школи. Зокрема, в 1–6-их класах комп'ютер може використовуватися як засіб навчальної діяльності з метою оволодіння школярами первинними вміннями і навичками роботи з ним; у 7–9-их класах забезпечується вивчення базового курсу «Основи інформатики», у 10–12-их класах — поглиблене вивчення окремих розділів інформатики з урахуванням профільності підготовки та за вибором учнів. Запроваджуються центри ресурсного програмного забезпечення всіх навчальних предметів».

Вивчення інформатики у Державному стандарті [2] віднесено до освітньої галузі «Технології». Зміст Державного стандарту щодо інформатики в основній і старшій школі розкритий у п'ятому пункті «Людина та інформаційна діяльність» і в цілому відповідає концепції [1]. Він передбачає вивчення основ інформатики (що позитивно) в основній школі, але включає сюди (чого ми не підтримуємо) питання моделювання, алгоритмізації і програмування. Зміст інформатики для старшої школи в проекті достатньо не конкретизований. Оскільки стандартні для інформатики теми відійшли до основної школи, він штучно доповнений темами, які не можна вважати стандартними: інформаційні технології в проектній діяльності, інформаційне забезпечення проектної діяльності тощо. Поруч знаходяться теми, які є ще менш значимими: системи штучного інтелекту, експертні системи. Ці теми у масовій школі прийнято вивчати оглядово, без глибоко занурення у відповідні системи, без розгляду мов логічного програмування тощо, оскільки вони сьогодні не домінують серед сучасних інформаційних технологій.

Після затвердження Типових навчальних планів для загальноосвітніх навчальних закладів [3] виявилося, що на інформатику в цілому протягом трьох



років відведено одну годину в тиждень із восьми годин освітньої галузі «Технології» в основній школі і три–чотири години з шести у старшій. Оскільки погодинне планування визначає обсяг і зміст навчання, то маємо суттєве протиріччя між концепцією і Державним стандартом, з одного боку, і Типовими планами — з другого. Сподівання багатьох фахівців, що вивчення інформатики розпочнеться з 7-го класу виявилися марними. Таке вивчення розпочнеться в 9-му класі в обсязі однієї інваріантної години на тиждень. Для типової школи, на нашу думку, це цілком прийнятно. Більше того, підтримуючи всі починання вчителів-ентузіастів, скеровані на широке використання комп'ютера в навчальному процесі в школах першого і другого ступенів і на створення пропедевтичних програмних засобів для зацікавлених дітей різного віку, зокрема, багатообіцяючий проект «Сходинки до інформатики», зазначимо, що розпочинати вивчення інформатики як предмета в типовій школі доцільно власне з 9-го класу. Також варто реалізувати зазначений в концепції [1] чіткий поділ на вивчення базового курсу інформатики і поглиблене вивчення окремих розділів інформатики з урахуванням профільності підготовки та за вибором учнів.

Тому механічно, відповідно до Типових планів, зміщуючи початок вивчення інформатики на два роки, отримуємо, що базовий курс інформатики слід реалізувати в 9–11-их класах, а поглиблений — у 12-му класі. Такий підхід дає змогу побудувати справді цікавий курс.

Великий і потужний резерв — варіативна година. Як розпорядитися нею? Багато вчителів вважають, що одного уроку в тиждень для вивчення інформатики як предмета замало. І з цим треба погодитися. Однак є також багато вчителів, які просто не бажають брати другу годину, навіть коли дирекція школи її пропонує. Це відбувається тому, що статус предмета невисокий і вчителі не відчують зацікавленості учнів у його вивченні. Статус предмета визначається перш за все наявністю вступних випробувань у вищій навчальній закладі. Тому, як тільки інформатику введуть до переліку предметів для вступних випробувань, що намагаються зробити деякі вищі навчальні заклади,

чи, точніше, до переліку предметів, з яких відбувається зовнішнє тестування, інтерес до вивчення предмета зростає. Зацікавленість до предмета багато учнів втрачають також після першого ознайомлення зі шкільними підручниками, які хронічно є не якісними, про що знають всі — від учителя до міністра.

Розглянемо, як найкраще розпорядитися варіативною годиною, якщо така година буде надана. У базовому курсі інформатики, тобто в 9–11-их класах, друга година дасть змогу більш системно і повно освітити теми, вивчення яких інколи перетворюється на шалену гонитву. Адже проекти програм, які будуть розглянуті нижче і які розраховані на одну годину в тиждень, є частково нереальними і популістськими. У 9-му класі за рахунок варіативної години можна було би добитися віртуозного володіння клавіатурою, у повному обсязі вивчити теми «Текстовий редактор», «Презентації», які частково чи повністю з незрозумілих причин у проектах віднесені до 10-го чи навіть 11-го класу. В 10–11-их класах була б змога повністю завершити вивчення базового курсу інформатики в рамках наведених нижче схем. Вивчати базові теми інформатики в 12-му класі немає сенсу. Головне завдання учня на цьому етапі — застосовувати отримані раніше знання, зокрема, під час вивчення інших предметів чи підготовки до випускних іспитів (тестування). Тому роль варіативної години для 12-го класу зменшується. Дивно, але Типові плани надають дві інваріантні години для чотирьох профілів природничо-математичного напрямку власне у 12-му класі. На нашу думку, у 12-му класі заняття з інформатики варто проводити виключно в рамках профільного навчання з вивченням курсів за вибором.

Зробимо висновок з проблеми суперечності нормативних документів. Будь-яка навчальна програма вивчення базового курсу інформатики, а також проект підручника, або не відповідатиме Державному стандарту, або не відповідатиме Типовому навчальному плану.

Розглянемо проект базової програми з інформатики для 9–12-их класів, який переміг у конкурсі 2004 року [4]. Програма була доступна на сайті МОНУ лише короткий час і через це з нею знайомі далеко не всі. Структура програми така:

9-й клас (34 год.)

1. Інформація та інформаційні процеси (6 год.).
2. Інформаційна система (4 год.).
3. Операційні системи (7 год.).
4. Програмні засоби загального призначення (9 год.).
 - 4.1. Графічний растровий редактор (3 год.).
 - 4.2. Текстовий редактор (6 год.).
5. Програмні засоби навчального призначення (4 год.).
6. Глобальна мережа Інтернет. Пошук відомостей в Інтернеті (4 год.).

10-й клас (34 год.)

7. Програмні засоби загального призначення (8 год.).
 - 7.1. Графічний векторний редактор (3 год.).
 - 7.2. Текстовий редактор (5 год.).
8. Комп'ютерні презентації (7 год.).

9. Електронні таблиці (8 год.).

10. Програмні засоби навчального призначення (5 год.).

11. Електронна пошта. Форуми. Засоби спілкування в Інтернеті (6 год.).

11-й клас (34 год.)

12. Підготовка публікацій (5 год.).

13. Електронні таблиці (5 год.).

14. Бази даних. Експертні системи (11 год.).

15. Розширений пошук відомостей в Інтернеті. Безпека в Інтернеті (4 год.).

16. Засоби створення гіпертекстових сторінок (5 год.).

17. Програмні засоби навчального призначення (4 год.).

12-й клас (34 год.)

18. Алгоритми і алгоритмічні структури (9 год.).

19. Проектування програм (1 год.).

20. Системи візуального програмування (24 год.).

Проект програми відповідає Типовому плану, але не відповідає концепції і Державному стандарту, оскільки вивчення теми «Алгоритмізація і програмування» винесено у 12-й клас. На науково-практичних конференціях і в методичних публікаціях не раз піднімалося й обґрунтовувалося питання про необхідність вивчення цієї теми раніше, на рівні 10-го класу. До недоліків програми слід віднести також те, що теми, що стосуються текстового редактора, пропонується вивчати у різних класах: в 9-му, 10-му і 11-му класах, з електронними таблицями — у 10-му і 11-му. Багато є запитань до змісту розділів 18–20. Програма не є достатньо чіткою, конкретною і стрункою. Забагато уваги надається застосуванню комп'ютера для вивчення інших предметів, що не є питанням власне інформатики.

Інший проект задекларований як проект профільної програми для суспільно-гуманітарних та інших напрямків рівня стандарту для 10–12-их класів [5]. Передбачається, що в 9-му класі навчання відбуватиметься за попередньою програмою [4]. Особливості профільного навчання в проекті не розкриті. Порівнюючи з [4], бачимо, що тут механічно зменшено кількість годин на вивчення теми «Алгоритмізація і програмування» з 34-х до 19-ти і немає теми «Підготовка публікацій». Деяко збільшена кількість годин на вивчення мережевих технологій і ще деяких незначних тем. Програма має таку структуру:

10-й клас (30 год.+4 год. резерв)

1. Основи програмування. Візуальне середовище програмування (7 год.).
2. Основи інформаційної безпеки (4 год.).
3. Електронна пошта (4 год.).
4. Обробка текстових документів. Текстовий процесор (6 год.).
5. Векторна комп'ютерна графіка (4 год.).
6. Електронні таблиці (6 год.).

11-й клас (30 год.+4 год. резерв)

7. Основи структурного програмування (12 год.).
8. Спілкування в Інтернеті (4 год.).

9. Програмні засоби навчального призначення (3 год.).

10. Комп'ютерні мережі і телекомунікації (3 год.).

11. Комп'ютерні презентації і мультимедіа (8 год.).

12-й клас (31 год.+3 год. резерв)

12. Налаштування операційної системи. Сервіси (3 год.).

13. Аналіз даних у середовищі табличного процесора (5 год.).

14. Бази даних та експертні системи (10 год.).

15. Створення веб-ресурсів (7 год.).

16. Інтегроване використання засобів обробки документів (2 год.).

17. Спільна робота з документами. Розробка колективного проекту (4 год.).

Програма має суттєві недоліки. Вона не забезпечує неперервності в навчанні. Темі 1 і 7 необґрунтовано роз'єднані. Це ж стосується тем 6 і 13; 3, 8 і 10. Не на місці розташовані теми 2 і 11, оскільки презентації вивчати в 11-му класі запізно. Послідовність тем для 10-го і 11-го класу нелогічна. Поручена добра традиція: теми, що стосуються Інтернету, розглядати в кінці навчального року, що дає змогу маневрувати з резервом часу. Програма є частково нереальною. Для цього достатньо навести один приклад. У пункті 16 пропонується вивчати тему про створення однотипних листів масової розсилки із застосуванням технології злиття документів. Дана тема є складною навіть для студентів, тому що відповідне завдання вони виконують не швидше, ніж за 45 хвилин. Не ясно, чому ця тема пропонується для школи на рівні стандарту. Подібною є тема, що стосується створення й адміністрування форумів. Це також далеко не рівень стандарту.

Не зрозуміло також, як два наведені вище проекти програм узгоджуються між собою. Вони демонструють два різних погляди на одні й ті ж питання, що не мають жодних переваг один над одним.

Кожний проект відображає особисті погляди авторів на викладання інформатики і зміст навчання, тобто є суб'єктивним. Об'єктивну програму можна створити лише з урахуванням поглядів, побажань і рекомендацій широкого кола фахівців. Тому важливою і ледве не єдиною перевагою проектів є те, що автори проектів програм надають право вчителям змінювати порядок вивчення тем і перерозподіляти кількості годин, які відводяться для вивчення цієї чи іншої теми. Очевидно, це слід буде активно використовувати як в межах навчального року, так і в межах усього курсу в старшій школі.

Розглянемо ще один проект програми [6]. Це програма профільного вивчення інформатики в 10–12-их класах для фізико-математичного і екологічного профілів природничо-математичного напрямку, для яких Типові плани чомусь надають дві інваріантні години у 12-му класі. Програма створена шляхом механічного розширення програми [5] в частині, що стосується 12-го класу, двома темами з метою заповнення додаткових годин навчання: «Алгоритми і структури даних» (18 год.) і «Програмні засоби для математика»

(4 год.). Зміст цих тем описаний непереконливо. Зміст першої теми більше відповідає інформаційно-технологічному профілю, а другій темі не надано належної уваги. Щодо змісту інших розділів, то два останні проекти програм не відрізняються між собою. Це досить дивно, оскільки ще раз підкреслює, що проекти не забезпечують реальної профільності навчання за єдиним непереконливим винятком, що стосується згаданих двох профілів із тридцяти. Через це проблему профільного вивчення інформатики в типовій 12-річній школі проекти [5, 6] не розв'язують.

У проектах програм не враховані особливості економічного профілю суспільно-гуманітарного напрямку. Цей профіль користуватиметься підвищеною популярністю. Інформатика тут вимагає зовсім іншого наповнення, ніж інші профілі. На жаль, ні Типові плани, ні проекти програм не враховують особливостей вивчення інформатики в межах цього перспективного профілю.

Скористаємось наданим правом щодо зміни порядку слідування тем, врахуємо вимоги концепції [1] і Державного стандарту [2] і запропонуємо в рамках Типових планів [3] логічно завершену схему вивчення інформатики в типовій школі з урахуванням наведених вище проектів програм [4–6].

9-й клас

1. Теоретичні основи інформатики: інформація та інформаційні процеси.

2. Операційні системи.

3. Графічний (растровий) і текстовий редактори. Створення презентацій.

4. Інтернет: пошук інформації та електронна пошта.

10-й клас

5. Теоретичні основи інформатики: булева алгебра тощо.

6. Моделювання, алгоритмізація, процедурне і візуальне програмування.

7. Векторна графіка. Підготовка публікацій і презентацій у вигляді проектів.

8. Інтернет: теоретичні аспекти мереж і телекомунікації, засоби спілкування, мережевий етикет тощо.

11-й клас

9. Теоретичні основи інформатики: кодування інформації і системи числення тощо.

10. Електронні таблиці.

11. Бази даних і експертні системи.

12. Інтернет: створення сайтів, питання безпеки, авторське право.

12-й клас

13. Вивчення декількох тем за вибором залежно від профілю підготовки.

Наведемо список тем курсів за вибором, якими можна зацікавити учнів 12-го класу різних напрямків підготовки:

1) офісні технології;

2) операційні системи;

- 3) видавничча справа;
- 4) комп'ютерна графіка;
- 5) веб-дизайн і веб-програмування;
- 6) візуальне програмування у Visual Basic .NET;
- 7) програмування на C#;
- 8) програмування на Java;
- 9) бухгалтерські та інші програми для бізнесу;
- 10) засоби штучного інтелекту і експертні системи;
- 11) технології проектування (САПР);
- 12) основи комп'ютерної безпеки;
- 13) сучасні сервіси Інтернету;
- 14) інформаційна безпека;
- 15) комп'ютерна лінгвістика;
- 16) фото і відео монтаж;
- 17) автоматизація математичних обчислень;
- 18) інші інформаційні технології.

Профільне навчання буде підтримуватись серією навчальних посібників для курсів за вибором та для позакласного читання. Деякі з них уже випущені видавництвом ВНУ [7–9]. Деякі готуються до друку. Інші чекають на своїх авторів, зокрема, у видавництвах «Аспект» і «Геол.» – «СПД Глинський», у яких також є достатньо напрацювань для таких курсів, наприклад, [10–12] тощо. Правда, тут виникає серйозне запитання: чи можуть курси за вибором повністю замінити профільне навчання? На нашу думку, ні. Така підміна є штучною, але на даному етапі з нею можна погодитися.

Розглянемо, як можна реалізувати більш реальне профільне навчання в рамках запропонованої вище схеми. Виокремимо два напрями: 1) суспільно-гуманітарний і 2) природничо-математичний. Інформаційно-технологічний профіль технологічного напрямку тут не розглядаємо, оскільки він не є масовим.

Для суспільно-гуманітарного напрямку у 10-му класі варто зменшити кількість годин на вивчення тем 5 і 6 і збільшити кількість годин на вивчення тем 7 і 8 динамічно, тобто на розсуд учителя, з врахуванням декількох факторів: рівня зацікавленості учнів і суті профілю, пропедевтичної підготовки, поточного стану технічної бази, наявності методичних засобів тощо. Для природничо-математичного напрямку переважну кількість годин варто надати на вивчення тем 5 і 6, а теми 7 і 8 вивчати оглядово або самостійно.

В 11-му класі для суспільно-гуманітарного профілю слід зменшити кількість годин на вивчення теми 9 і збільшити для теми 12. Теми 10 і 11 слід наповнити задачами з урахуванням профілю навчання. Такі задачі мають бути запропоновані в підручниках і посібниках.

У 12-му класі профільне навчання реалізується очевидним способом, тобто шляхом вибору декількох курсів із наведеного списку чи його розширення.

Таке тлумачення концепції вивчення інформатики в 12-річній школі узгоджується з нормативними документами [1–3], має оптимальну і чітку структуру, відповідає двом наведеним вище проектам програм з точністю до зміни порядку слідування тем і вже частково реалізоване в навчальній літературі [13–15] й апробоване. З відповідним проектом навчальної програми

можна ознайомитися на сайті www.hlinsky.lviv.ua й обговорити всі методичні питання на форумах, конференціях, семінарах тощо. Зауважимо, що зараз активним і корисним є форум «Всеукраїнське методоб'єднання учителів інформатики» за адресою informatics.at.ua/forum. У результаті обговорення згадана програма буде вдосконалюватись і подаватиметься на розгляд МОНУ як альтернативна у тому випадку, якщо проаналізовані вище проекти не будуть вдосконалені.

Варто пам'ятати, що через два-три роки комп'ютер буде доступний більшості сімей так, як зараз доступний телевізор. Комп'ютер прийде в кожний дім. Початкові, часто цілком достатні, навички роботи з ним більшість школярів отримуватимуть дома так, як це вже давно має місце в США тощо. В основній школі не буде необхідності вивчати інформатику на рівні користувача. Більшість школярів матимуть достатні навички щодо застосування комп'ютера в навчально-пізнавальній діяльності. Одне із завдань шкільної інформатики — систематизувати, узагальнити і розвинути їх у потрібному напрямі, доповнити і збагатити, орієнтуючись на цю більшість і надаючи змогу меншості швидко ліквідувати розрив. Інформатику в школі треба формувати як наукову дисципліну, нехай технологічну, використовуючи набутий на цей час досвід великої кількості методистів і фахівців.

Мета проведеного аналізу — загострити увагу методистів і авторів проектів програм на питаннях, які варто розв'язати на етапі доопрацювання проектів задля доведення їх до кращого концептуального стану, покращення структури і змісту шкільної інформатики, що дасть змогу створити повноцінне методичне забезпечення курсу підручниками і навчальними посібниками для якісного вивчення предмету у 12-річній школі на високому науково-методичному рівні.

Література

1. Концепція загальної середньої освіти (12-річна школа). Постанова Колегії МОНУ та Президії АПНУ № 12/5–2 від 22.11.2001 // Книга вчителя інформатики: Довідково-методичне видання. — Х.: Торсінг плюс, 2006. — 272 с.
2. Державний стандарт базової і повної середньої освіти. Постанова Кабінету Міністрів України від 14 січня 2004 р. № 24 // Книга вчителя інформатики: Довідково-методичне видання. — Х.: Торсінг плюс, 2006. — 272 с.
3. Про затвердження Типових навчальних планів загальноосвітніх навчальних закладів 12-річної школи. Наказ МОНУ від 9 березня 2005 р. № 145 // Книга вчителя інформатики: Довідково-методичне видання. — Х.: Торсінг плюс, 2006. — 272 с.
4. Жалдак М. І., Морзе Н. В. Програма курсу «Інформатика» для загальноосвітньої школи (9–12 класи) // Веб-ресурс МОНУ <http://www.mon.gov.ua/main.php?query=education/average>. — 2004 р.
5. Проекти навчальних програм для профільного навчання у 12-річній школі/Інформатика. 10–12 класи. Рівень стандарту. <http://www.mon.gov.ua/main.php?query=education/average/profil>.
6. Проекти навчальних програм для профільного навчання у 12-річній школі/Інформатика. 10–12 класи. Академічний рівень. <http://www.mon.gov.ua/main.php?query=education/average/profil>.
7. Основи веб-дизайну // Пасічник О. Г., Пасічник О. В., Стеценко І. В. — К.: ВНУ, 2008. — 336 с.
8. Основи Інтернету // Левченко О. М., Завадський І. О., Прокопенко Н. С. — К.: ВНУ, 2008. — 320 с.

9. Завадський І. О., Заболотний Р. І. Основи візуального програмування. — К.: BHV, 2008. — 272 с.
10. Бондаренко О. О. Інформатика. 9 клас. Visual Basic. — Шепетівка: Аспект, 2006. — 192 с.
11. Бейсик. Від Qbasic до Visual Basic .NET // Глинський Я. М., Анохін В. Є., Ряжська В. А. Навч. посіб. 6-те вид. — Львів: СПД Глинський, 2007. — 192 с.
12. Глинський Я. М., Ряжська В. А. Інтернет. Мережі, HTML і телекомунікації: Навч. посіб.: 4-те доп. вид. — Львів: СПД Глинський, 2007. — 192 с.
13. Глинський Я. М. Інформатика: 9 клас. Перший рік навчання. — Львів: СПД Глинський, 2007. — 200 с.
14. Глинський Я. М. Інформатика: 10–11 класи: Навч. посіб.: У 2 ч. — 5–7-ме вид. — Львів: СПД Глинський, 2005–2007. — Ч. 1. Алгоритмізація і програмування. — 256 с.
15. Глинський Я. М. Інформатика: 10–11 класи: Навч. посіб.: У 2 ч. — 5–7-ме вид. — Львів: СПД Глинський, 2005–2007. — Ч. 2. Інформаційні технології. — 264 с.

★ ★ ★

ПРОГРАМНА ІНЖЕНЕРІЯ — СПАДКОЄМИЦЯ ПРОГРАМУВАННЯ

Лавріщева К. М.

З появою перших електронних обчислювальних машин (ЕОМ) виникло програмування як процес побудови програм для розв'язування деяких задач на цих машинах.

Програма — це послідовність дій (команд машини, операторів мов програмування, сценаріїв тощо), яка відображає опис постановки задачі і хід їх виконання в комп'ютері. Спочатку опис програм був у командах ЕОМ і тому програмування розглядалося як мистецтво вистроювати необхідні команди машини під представлення конкретних цілей задач, що розв'язувалися на ЕОМ.

Одночасно створювалися і мови програмування, починаючи з простих, які відображали командний стиль цих ЕОМ. Наприклад, перша мова програмування, так звана, адресна мова (1957–58 р.р.), яку створено першою програмісткою СРСР професором К.Л. Ющенко у стінах Інституту кібернетики ім. академіка Глушкова Національної академії України. Інститут кібернетики, як раз у грудні 2007 р., святкувала своє 50-річчя, і я, як учениця К.Л. Ющенко (з 1959 р.), послідовниця її напрямку програмування у кібернетичному виміру віддаю їй пошану і пам'ять, вважаю її великою україankoю, яка багато зробила в становленні і розвитку програмування для забезпечення професійного використання комп'ютерів у різних сферах нашого буття.

Перші закордонні мови, що широко стали відомі в усьому інформаційному світі на той час (Алгол, Кобол, Фортран та інші), були орієнтовані на опис математичних і обчислювальних задач для їхнього розв'язування в ЕОМ. Опис програм цими мовами потребував розроблення систем програмування, які перетворювали мову опису програм на мову тої чи іншої ЕОМ, так звані транслятори, компілятори, інтерпретатори тощо.

Поступово у програмуванні за участю теоретиків та практиків складалися загальні погляди, концепції і підходи щодо опису й реалізації програм на ЕОМ.

Від редакції. У статті зроблена спроба наукового обґрунтування змісту шкільної інформатики.

Як завжди, автори Глинський Я. М. і Ряжська В. А. оригінальні у своїх підходах щодо цієї проблеми. Погодимось чи не погодимось з пропозиціями, думками та висновками авторів — це справа наших читачів. Звернемо тільки увагу на те, що у запропонованій схемі змісту інформатики у 9–12-их класах автори застосовують висловлення «теоретичні основи інформатики» тільки перед такими назвами: інформація та інформаційні процеси, булева алгебра, кодування інформації та систем числення. Можна подумати, що на цьому теорія ІКТ закінчується, що принципово є неправильним. На жаль, сучасний курс Інформатики справді орієнтований на формування тільки практичних умінь (роботи з офісними пакетами), а не на ознайомлення з основами теорії ІКТ.



Вони обґрунтовувалися і перевірялися практикою конкретного програмування. До них належать методи програмування (мова і спосіб її застосування для доказового опису змісту програм), а саме теоретичного, формального та прикладного напрямку.

Зараз настав час, коли програмування стало масовою діяльністю, в ній беруть участь спеціалісти, що пишуть різні комп'ютерні програми, більш 10 мільйонів і близько 100 мільйонів (статистика останніх трьох років), що їх використовують. Програмування як діяльність багатьох людей поступово переходило на технологічні напрямки, а саме створення процесів поступового розроблення складних програм, планування робіт у колективі й керування ними для успішного їх виконання й отримання відповідної оплати за свою працю, як продукту діяльності.

Становлення програмної інженерії та її призначення

Як результат багаторічної праці світового програмуючого загалу накопичилась велика кількість знань та досвід побудови різноманітних комп'ютерних програм. Усе це зумовило необхідність систематизувати набуті знання і визначити їх у вигляді дисципліни (предмету) з метою формування спільного бачення проблеми комп'ютерною спільнотою. Спеціально створений комітет спеціалістів з інформатики при ACM і IEEE Computer Society сформував базове ядро знань SWEBOOK (Software Engineering body of Knowledge — 2001 р.), у якому в концентрованому вигляді подав концептуальний зміст десятих базових галузей (knowledge areas) та дефініції програмної інженерії (ПІ) [1–3].

Ураховуючи й те, що ПІ входить до складу комп'ютерної науки (Computer science) як головна науково-

технічна дисципліна проектування комп'ютерних систем і нова дисципліна керованого розроблення комп'ютерних програм, вона стала в багатьох передових державах світу предметом навчання більш 15 років. Кафедри програмування в університетах на факультетах інформатики цих держав стали зватися програмною інженерією. В Україні перехід до навчання програмної інженерії було зроблено лише після Постанови Кабінету Міністрів України (від 13 грудня 2006 р. №1719) про затвердження нового переліку напрямків навчання, у якому програмна інженерія є розділом комп'ютерних наук за змістом — програмування програмних і автоматизованих систем (але не зовсім таким є її зміст). Але цим підкреслено початок навчання програмної інженерії у сфері освіти, хоча посібників і підручників щодо предмету навчання Програмної інженерії в Україні практично не має.

У зв'язку з цим, такий підручник було подано на конкурс «Вчитель-новатор» 2007 р. Короткий зміст цього підручника викладається у цій статті. Вона ставить своєю метою ознайомити викладачів навчальних закладів освіти з призначенням цієї дисципліни і підходом до її навчання. Автор має великий досвід у викладанні цього предмету на кафедрі МФТІ філії в Інституті кібернетики НАН України (2001–2007 рр.), а також на факультеті кібернетики у Київському Національному університеті імені Тараса Шевченка.

Наведемо спочатку визначення ПІ, що було дано у ядрі знань SWEBOOK, і нове, більш уточнене.

Визначення 1. Програмна інженерія (Software Engineering) — це система методів, способів і дисциплінарних заходів із планування, розробки, експлуатації і супроводу програмного забезпечення (ПЗ), придатних для масового виробництва. Ця інженерна дисципліна охоплює всі аспекти створення ПЗ від початку формулювання системних вимог, через розроблення продукту і до його використання, супроводження та остаточного списання.

На нашу думку, ця дефініція дещо звужує сутність не лише поняття предмету ПІ, але й об'єктів, що застосовуються у ПІ, тобто не тільки ПЗ, але й інших цільових об'єктів. Тому дається їхнє уточнення у більш широкому розумінні, долучаючи різні аспекти, що характеризують глибинні питання ПІ, як наукової та інженерної дисципліни.

Відомо, що будь-яка наука — це система перевірених практикою знань, які відображають загальні питання, поняття і закономірності їх розвитку. Вона встановлює зв'язки з іншими науками і впливає на їхній розвиток. Щодо програмної інженерії можна сказати, що вона інтегрувала в собі концепції, принципи математики та головні положення фундаментальних наук, а саме: теорії алгоритмів, математичної логіки, теорії множин, теорії доведень, теорії управління тощо (рис. 1).

Ці науки утворюють теоретичний базис програмної інженерії, необхідний для побудови абстракцій програм за їхніми базовими поняттями та принципами, що перелічені нижче за кожною з фундаментальних наук:

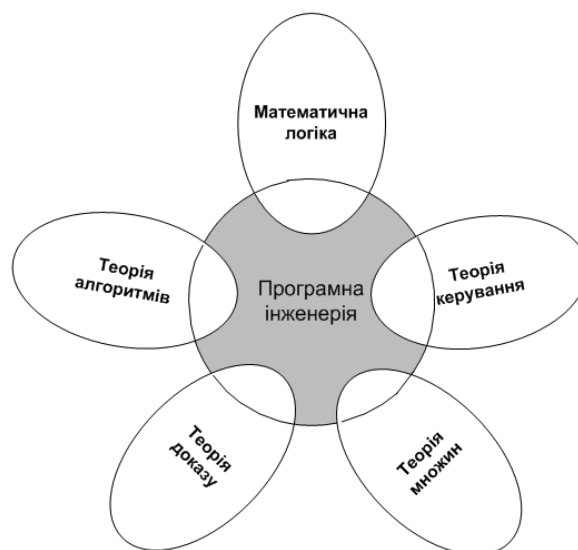


Рис. 1. Теоретичний фундамент програмної інженерії

- у теорії алгоритмів — нормальні алгоритми, обчислювальні функції, машина Тьюрінга, алгоритмічні алгебри, граф-схеми, моделі алгоритмів і програм тощо;
- у математичній логіці — логічні числення і висловлювання теоретичного створення правильних умововиведень та логіко-алгебраїчний апарат специфікації програм;
- у теорії доведень — математичне доведення за аксіомами і твердженнями програм, виведення теорем, теорія верифікації програм, теорія надійності ПЗ;
- у теорії множин — квантори загальності, існування, операції над множинами, що застосовуються для формального подання аксіом про різні сукупності створювання складних програмних об'єктів;
- у теорії керування — принципи, методи та загальні закони планування і керування процесами отримання й оброблення інформації в кібернетичних й управлінських системах.

Крім цього фундаменту до системи знань програмної інженерії входить все те, що накопичено в програмуванні:

- формальні методи програмування — специфікація програм, їх доведення, верифікація і тестування, а також математичні моделі надійності, ризику тощо;
- прикладні методи, а саме прийоми, принципи, правила, окремі дії і цілісні процеси життєвого циклу (ЖЦ) виробництва комп'ютерних систем, що є інструментами колективної розробки, застосовуваними виконавцями великих програмних проектів;
- методи керування колективами, а саме планування за мережевими графіками, контролювання робіт у процесах ЖЦ, вимірювання і оцінювання якості проміжних результатів виробництва, прогнозування і регулювання строків і вартості виготовлення продукту, а також його сертифікації [5, 6].

Тобто програмна інженерія, як спадкоємиця науки програмування, залучила до сфери свого поширення всі теоретичні і прикладні досягнення, набуті за пе-

рід її існування, і, таким чином, програмна інженерія склалася як науково-інженерна дисципліна.

Тому природно будемо вважати визначення 1, наведене в ядрі знань SWEBOOK, вузьким, і дамо нове визначення програмної інженерії з точки зору наукової та інженерної дисципліни.

Визначення 2. Програмна інженерія — розділ комп'ютерної науки, який вивчає методи і засоби побудови комп'ютерних програм; відображає закономірності розвитку та узагальнює накопичений досвід програмування; оперує об'єктами (модулями, компонентами, програмними аспектами тощо) та визначає автоматизовані операції щодо їх вироблення; виробляє правила і порядок інженерної діяльності і керування технологічним процесом побудови з простих об'єктів нових, більш складних об'єктів (програмного забезпечення, програмних систем (ПС), сімейств систем, програмних проєктів тощо), а також методи вимірювання й оцінювання готового продукту.

Це тлумачення програмної інженерії подано в більш широкому сенсі, а саме з позиції застосування в ній теорії програмування та інженерії вироблення програмних продуктів, яка сформувалася шляхом адаптації загальних методів керування (наприклад, метод критичного шляху, PERT, за операціями), яким властиво розподіл робіт між різними виконавцями проєкту, оцінювання трудовитрат на кожен роботу, вартості виготовлення кінцевого продукту та його показників (якості, ефективності, точності тощо).

Таким чином, дисципліна програмна інженерія — це наукова та інженерна дисципліни вироблення програмних продуктів (рис. 2).

На перетині областей визначення III (овалів на рис. 2) — теорія та практика побудови складних програмних об'єктів. Теорія побудови — це теорія їх програмування за абстрактними специфікаціями (графовими і структурними схемами, функціями і композиціями, дескрипторами і номінативними даними, сценаріями використання — use case діаграми тощо), а також формальна перевірка відповідності цільових об'єктів специфікаціям за методами доведення, верифікації, інспекції тощо. Практика побудови — це застосування теоретичних і практичних методів інженерії програмування шляхом використання засобів перевірки (верифікації, валідації, тестування) специфікацій об'єктів, інструментів їхнього послідовного трансформування до результуючого коду та інженерія оцінювання і сертифікації різних показників якості виготовленого програмного продукту.

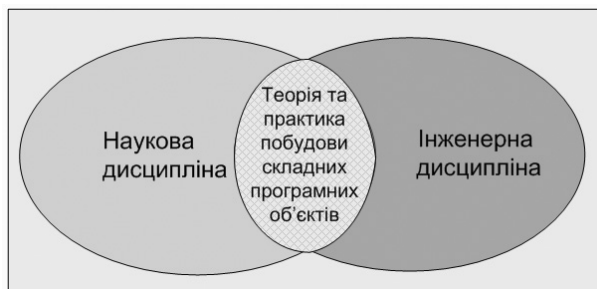


Рис. 2. Наукова та інженерна дисципліни програмної інженерії

Визначення програмної інженерії як наукової дисципліни

На відміну від математичної або інших фундаментальних наук, метою яких є отримання нових знань для розв'язання відповідних задач, метою програмної інженерії є застосування знань у розробці складних програмних об'єктів, де знання є уособленням загальної теорії побудови програм для комп'ютерів, орієнтованої на виготовлення продукту, впровадження якого принесе певну користь споживачеві.

Програмна інженерія як наукова дисципліна охоплює теоретичні, формальні методи та відповідні засоби побудови складних програмних об'єктів. Побудова має на меті аналіз предметної області, що автоматизується, і продукування результуючого коду для виконання на комп'ютері. Інтегровані в програмну інженерію фундаментальні науки, згадані вище (див. рис. 1), а також наука програмування складають її загальний теоретичний фундамент, який подає базові поняття, об'єкти і формальні механізми, необхідні для надання програмним продуктам загальних властивостей та специфічних рис відповідно до встановлених до них вимог. III як наука включає:

- 1) основні поняття і об'єкти;
- 2) теорію програмування і керування виготовленням програмного продукту;
- 3) засоби й інструменти процесів практичного розроблення продукту.

1. Основні поняття програмної інженерії це: дані і їхні структури (прості і складні), функції і композиції, базові об'єкти (модуль, об'єкт, компонент, каркас, контейнер, повторно використовуваний компонент (ПВК) тощо) і цільові об'єкти (програмне забезпечення, програмна система, сімейство систем, програмний проєкт, складні програмні застосування тощо).

Розроблення простих об'єктів виконується через елементарні дії щодо їхнього формального опису, а розроблення цільових об'єктів — через застосування інженерних методів, включаючи керування строками і вартістю виробництва.

Дамо визначення цільових об'єктів, у тому числі головного об'єкта — ПЗ.

Програмна система (Application) — комплекс інтегрованих програм і засобів, що реалізують набір взаємопов'язаних функцій деякої предметної галузі в заданому середовищі. У комплекс можуть входити: прикладні системи (наприклад, програми розрахунку зарплати, обліку матеріалів на складі тощо), загальносистемні програмні засоби (наприклад, транслятор, редактор, СКБД і т. п.), спеціалізовані програмні засоби для реалізації функцій захисту інформації, забезпечення безпеки функціонування та ін. Спосіб виготовлення — інженерія ПС (або application engineering), що включає процеси ЖЦ, методи розробки і процедури керування, а також методи і засоби оцінювання продуктів і процесів з метою їх удосконалення.

Програмне забезпечення — сукупність програмних засобів, які реалізують функції комп'ютерної системи (або функції технічної апаратно-програмної системи), включаючи загальносистемні засоби (напри-

клад, ОС, СКБД, вбудовані підсистеми контролю показників технологічних процесів, оброблення сигналів тощо) та прикладні програмні системи. Так, функціями деякої ОС є керування задачами, програмами, даними і т. п. Вона може входити до складу ПС або бути ідентичною функціям програмної системи. Спосіб виготовлення — інженерія розроблення цільових програм для задач стосовно ПЗ.

Сімейство систем (systems family) — сукупність програмних систем із загальним (незмінним для всіх членів сімейства) і керованим (змінним) набором характеристик, що задовольняють визначеним потребам прикладної області (домену). Спосіб виготовлення — інженерія домену (Domain Engineering) або конвеєрне виробництво однотипних ПП за єдиною схемою на основі спеціально розроблених базових членів сімейства й інших готових програмних активів (assets) за допомогою базового процесу або автоматизованої лінійки продукту (Product line).

Програмний проект — унікальний і інтегрований комплекс взаємозалежних заходів, орієнтованих на досягнення цілей і задач об'єкта розроблення за визначеними вимогами до строків, бюджету та характеристик очікуваних результатів діяльності від нього. Спосіб виготовлення — інженерія процесу розроблення і менеджменту проекту.

Складні програмні об'єкти — сукупність взаємопов'язаних цільових об'єктів різних типів, які виконують необхідні функції в складній системі, подані як самостійно розроблені прості та цільові об'єкти або вибрані з репозитарію готових ресурсів.

2. Теорія програмування — сукупність методів, мов і засобів опису та проектування цільових об'єктів, а також методів їх доведення, верифікації і тестування [6–8] на правильність їх створення. Поряд з об'єктами теорії програмування до програмної інженерії залучені формальні методи керування проектом (персоналом, матеріальними та фінансовими ресурсами) і його окремими характеристиками (рис. 3).

З теорії програмування в ПІ подані такі методи:

- методи теоретичного і прикладного програмування (алгебраїчний, об'єктний, компонентний тощо), призначені для проектування різних типів цільових об'єктів;
- методи перевірки правильності за формальними процедурами (твердження, виведення, доведення);

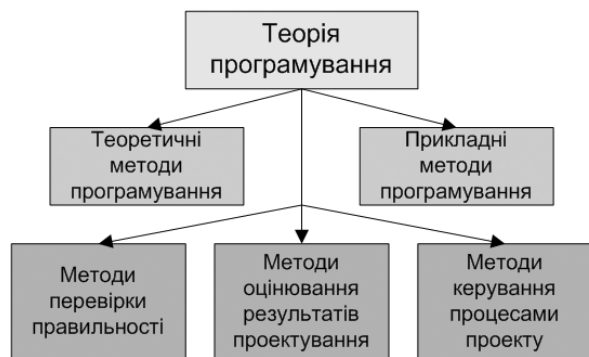


Рис. 3. Сукупність методів програмної інженерії

- методи оцінювання результатів послідовного проектування (проміжних робочих продуктів) і кінцевого продукту відносно встановлених показників (надійність, якість, точність, продуктивність тощо);
- методи керування і контролю розроблення продукту проектів, а також розрахункові методи (трудозатрат кожного розробника, вартості робіт тощо).

3. Засоби та інструменти ПІ. До об'єктів, що є артефактами поступового створюваного ПП, включаються різного роду описи: вимоги до розробки ПП, погоджені із замовником, архітектура, структури даних, специфікації програм і т. п.

Проектування об'єктів виконується за допомогою сучасних візуальних мов, наприклад UML, мов програмування (C++, Basic, Java, Pascal тощо) з використанням відповідних інструментальних середовищ, що містять необхідні мовні перетворювачі й інструменти підтримки різних артефактів ПП, що розробляються. Як засоби їх подання застосовуються діаграми використання, потоків даних, класів, поведінки, а також шаблони, каркаси, темплейти тощо.

Перевірка правильності цільових об'єктів здійснюється за допомогою вказаних методів і відповідних інструментів, пристосованих до цілей розроблення різних задач проекту у середовищі проектування. Готовий продукт перевіряється на відповідність реалізованих функцій заданим вимогам, тестується за спеціальними методиками, а також піддається вимірюванню та оцінюванню з метою отримання особливих показників, у тому числі якості.

Проектування цільових об'єктів виконується у сучасному середовищі (наприклад, Microsoft Visual Studio, MSF, Rational Rose тощо), за допомогою не тільки інструментів перетворення описів різних типів цільових об'єктів, а й на засобах керування проектом (наприклад, Project Microsoft), зокрема персоналом, планами і різними ресурсами у проекті.

Визначення програмної інженерії як інженерної дисципліни

Програмна інженерія як інженерна дисципліна (або власне інженерія) — це сукупність прийомів виконання діяльності, пов'язаної з виготовленням програмного продукту для різних видів цільових об'єктів із застосуванням методів, засобів та інструментів наукової складової програмної інженерії [6–8]. Основу інженерії продукту складають базові елементи процесу виготовлення програмного продукту (рис. 4):

1) ядро знань SWEBOOK як набір теоретичних концепцій і формальних визначень стосовно методів і засобів розроблення та керування програмними про-

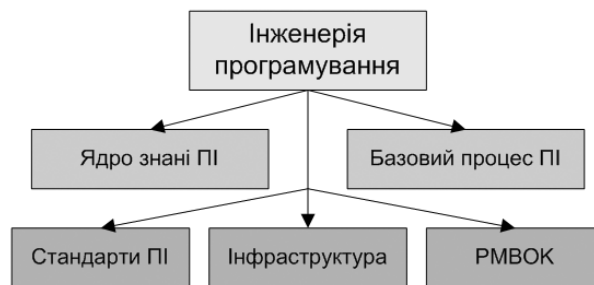


Рис. 4. Базові складові інженерної дисципліни

ектами, які можуть застосовуватися в інженерії програмування;

2) базовий процес ПІ, як стрижень процесної діяльності в організації-розробника програмного продукту;

3) стандарти як набір регламентованих правил конструювання проміжних артефактів у процесах ЖЦІ;

4) інфраструктура — умови середовища та методичне забезпечення базового процесу ПІ і підтримка дій його виконавців, що займаються виробленням програмного продукту;

5) менеджмент проекту (РМВОК) — ядро знань з керування промисловими проектами, яке містить набір стандартних процесів, а також принципів і методів планування і контролювання роботами в проекті.

Інженерну дисципліну можна розглядати з двох пов'язаних точок зору:

- як инженерну діяльність, де інженери різних категорій виконують роботи в рамках проекту, використовуючи відповідні теоретичні методи і засоби ПІ, що рекомендовані в ядрі знань SWEBOOK, а також стандарти процесів проектування цільових об'єктів за обраними методами;
- як систему керування проектом, якістю і ризиками із залученням правил і положень стандартів ЖЦІ, якості та менеджменту проекту.

Інженерна діяльність повинна плануватися з обов'язковим розподілом робіт у проекті між різними категоріями виконавців. Менеджер проекту — це головна дійова особа проекту, відповідальна за проектування і контроль виконання робіт спеціальними службами інфраструктури проекту в організації, зокрема службами верифікації, тестування, якості тощо. Продукт колективного виготовлення передається замовнику для супроводу. У ньому можуть бути знайдені різні помилки і недоліки, які усувають розробники.

Ця діяльність у програмній інженерії практично вже відпрацьована і за своєю сутністю близька до инженерної діяльності у промисловості, де інженерія — це спосіб застосування наукових результатів у виготовленні технічних виробів на основі технологічних правил і процедур, методик вимірювання, оцінки і сертифікації щоб задовольнити й отримати користь від виготовленого продукту або товару.

Далі дається змістовний розгляд складових инженерної дисципліни.

1. Ядро знань SWEBOOK — стислий опис концептуальних основ програмної інженерії. Структурно поділяється на 10 розділів (knowledge areas), які умовно можна розкласти на дві категорії: проектування продукту й инженерну діяльність щодо його вироблення. Перша категорія — це методи і засоби розробки (формування вимог, проектування, конструювання, тестування, супровід), друга категорія — методи керування проектом, конфігурацією і якістю та базовим процесом організації-розробника.

Наповнення базового процесу програмної інженерії методами з ядра знань SWEBOOK, а також задачами і діями стандартного ЖЦІ, обумовлює його пристосування до потреб конкретної організації-розробника щодо певної регламентованої послідов-

ності розробки і супроводу програмного продукту. Усе це створює технологічний базис інженерії виготовлення конкретного продукту (або низки однотипних продуктів) в організації або колективу, групи, маленької фірми тощо.

2. Базовий процес (БП) — це метарівень для забезпечення «процесного продукування» продукту. Він містить основні поняття стосовно оснастки, організаційної структури колективу розроблювачів та методології оцінки, вимірювання, керування змінами й удосконалювання самого процесу. У цілому базовий процес містить множину логічно пов'язаних з ним видів инженерної діяльності організації-розробника та набір засобів та інструментів щодо виготовлення програмного продукту.

3. Інфраструктура — це набір технічних, технологічних, програмних (методичних) та людських ресурсів організації-розробника, необхідних для виконання підпроцесів базового процесу програмної інженерії, орієнтованого на виконання договору із замовником програмного проекту. До технічних ресурсів належать: комп'ютери, пристрої (принтери, сканери тощо), сервери і т. п. До програмних — загальносистемне ПЗ середовища розробки, напрацювання колективу, оформлені у вигляді повторно використовуваних компонентів та інформаційне забезпечення. Технологічні та методичні ресурси включають методики, процедури, правила, рекомендації стандартів щодо процесу і керування персоналом, включаючи комплект документів, що встановлює регламент виконання і регулювання процесів ЖЦІ, пристосованих для вирішення конкретних задач проекту. Людські ресурси — це групи розробників та служб керування проектом, планами, якістю, ризиком, конфігурацією та перевірки правильності виконання проекту розробниками (рис. 5).

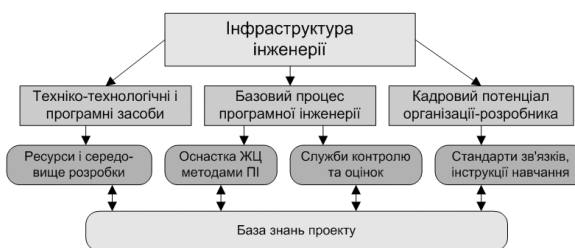


Рис. 5. Загальна інфраструктура проекту

Після виконання проекту й отримання досвіду побудови конкретного продукту, базовий процес і його окремі елементи, подані на рис. 5, можуть удосконалюватися (через доопрацювання або зміни прийомів розроблення або перевірки, виконання доробок, змінювання, додавання нових засобів) відповідно до вимог стандарту ДСТУ ISO/IEC 15504-7 (Оцінювання процесів ЖЦІ ПЗ. Наставови з удосконалення процесу) з метою підвищення рівня можливостей і оцінки потужності базового процесу.

4. Стандарти ПІ — встановлюють технологічно відпрацьований набір процесів зі строго визначеним і регламентованим порядком проведення різних видів робіт у програмній інженерії, пов'язаних із розробленням програмного продукту й оцінюванням його якості, ри-

зику тощо. Стандарти у галузі програмної інженерії регламентують різні напрямки діяльності щодо програмування програмних продуктів. Вони стандартизують термінологію і поняття, життєвий цикл, якість, вимірювання, оцінювання продуктів і процесів. Найбільш важливими серед них є стандарт ISO/IEC 12207 «Процеси життєвого циклу програмного забезпечення», ДСТУ ISO 9001 «Системи управління якістю. Вимоги», ДСТУ 2844–94, ДСТУ 2850–94, що регламентують різні інші аспекти забезпечення якості ПП.

Процеси ЖЦ в стандарті ISO/IEC 12207 подають загальні положення, задачі і регламентовані дії з проектування, а також рекомендації щодо застосування цих процесів для розроблення і контролю проміжних результатів. У стандарті містяться також організаційні процеси — планування, керування і супровід. Процес планування призначено для складання планів, графіків робіт щодо виконання проекту і розподілу робіт між різними категоріями фахівців, а також для контролю планів і виконаних робіт. Процес керування проектом визначає задачі та дії з керування роботами у проекті, що виконуються фахівцями, які володіють теорією управління, а також стеження за плановими строками, що надані замовником проекту. Процес супроводу — включає дії щодо покращення готового продукту, виявлення й усунення знайдених в ньому недоліків і внесення нових або видалення деяких функцій у продукті.

5. Менеджмент проекту — це керування розробленням проекту з використанням теорії управління та процесів ядра знань РМВОК (Project Management body of knowledge) [9]. У настанові з використання РМВОК подано положення і правила керування часовим виробничим циклом побудови унікального продукту в рамках проекту. РМВОК є стандартом, розробленим американським Інститутом управління проектами (www.pmi.org). Ядро знань РМВОК містить опис лексики, структури процесів і пов'язаних з ними галузей знань. У цілому воно відображає сучасну практику керування проектами в різних областях промисловості. Головним областям дається трактовка стосовно програмного продукту.

Область знань керування вмістом проекту — це процеси, які необхідні для виконання робіт за проектом (виключаючи надлишкові дії), а також для його планування з розбивкою робіт на простіші для спрощення процесу керування. Область керування якістю — це процеси й операції досягнення цілей проекту щодо якості продукту, правила і процедури для поліпшення процесу досягнення цілей проекту із забезпеченням якості відповідно заданим вимогам, а також процедури контролю результату на відповідність загальним стандартам якості. Область керування людськими ресурсами організації містить порядок розподілу робіт між виконавцями відповідно до їхньої кваліфікації і професіоналізму, а також процедури регламентування робіт щодо розроблення програмного продукту.

Таким чином, стандарт РМВОК, як і стандарт ISO/IEC 12207, мають багато спільного, особливо в плані організаційних процесів з керування проектом та інженерією доменів. Ядро знань SWEBOK і стан-

дарту ЖЦ також мають взаємозв'язки. Процесам ЖЦ зіставляються необхідні методи ядра і тим самим визначається базовий процес проекту. Він доповнюється методиками й обмеженнями щодо деяких особливостей вироблення продукту. Ядро знань SWEBOK і РМВОК пов'язані подібними моделями ЖЦ, методами та інструментами в частині керування процесами виконання проекту.

Тобто проведений розгляд питань вироблення програмних продуктів на процесній і інженерній основі дозволив сформулювати змістовний трикутник фундаменту програмної інженерії (рис. 6).



Рис. 6. Три кити програмної інженерії

У ньому головні елементи програмної інженерії (SWEBOK, СТАНДАРТИ, РМВОК) пов'язані між собою, доповнюють один одного і слугують головним цілям програмної інженерії щодо вироблення програмного продукту.

Практичне застосування предмету пі

Усі розглянуті базові теорії і складові елементи ПІ мають прикладне значення для застосування на практиці, як теорії та основних положень технології проектування програмних продуктів у певному середовищі розроблення. На жаль, в Україні немає своїх програмних середовищ для практичного розроблення програмних продуктів на інструментальній основі. Цю прогалину заповнюють закордонні інструментальні середовища та системи підтримки програмування, а саме системи програмування з різних сучасних мов (C++, C#, Basic, Paskal тощо). Наприклад, система Express C#, Basic. Вона дає можливості налагодити певну програму в цих мовах до кінцевого результату.

Прикладом реалізації інженерної дисципліни колективного проектування цільових об'єктів є система Visual Studio Teams Systems фірми Microsoft. У ній реалізовано технологію проектування, кодування, тестування та впровадження проектів. У середовищі системи реалізовано технологію підбору, розподілу і виконання робіт між різними групами спеціалістів програмного проекту. Вони природно мають різні рівні знань і навичок у галузі програмування і тому поділяються на чотири категорії.

Кожний спеціаліст із цих категорій отримує роботу відповідно до своїх здібностей, починаючи з підготовчої категорії. Перехід в іншу більш високу кваліфікаційну категорію залежить від якості виконання спеціалістом попереднього завдання й отриманого ним досвіду, як в частині знання проблематики і методів їхнього вирішення, так і самого пакета інструментів фі-

рми. Спеціалісти, що входять до четвертої категорії, несуть відповідальність за правильність розроблення продукту в цілому в певному середовищі. Фірма Microsoft впровадила цю систему в деяких університетах України, тому у студентів є можливість брати участь у розробленні пілотних проектів і тим самим отримувати практичний досвід застосування базових положень програмної інженерії та цієї системи.

Про підручник із програмної інженерії і програму її навчання

Про підручник. У поданому на конкурс «Вчитель-новатор» 2007 року в підручнику «Програмна інженерія» висвітлено всі питання наукової та інженерної дисципліни проектування комп'ютерних програмних систем. Системно викладено методи програмування, методи доведень і верифікації та тестування програм. Охарактеризовано базові елементи й інструменти інженерії розроблення цільових об'єктів ПІ (ПЗ, програмних систем, доменів, а саме проектів). У розділах, присвячених інженерній дисципліні виготовлення програмних продуктів, системно викладено усі аспекти програмної інженерії, які пов'язані з процесами ЖЦ, керуванням колективами виконавців, їхніми роботами, якістю виконання і вкладання у визначені строки. Висвітлено зміст і призначення стандартів ЖЦ і якості програмних продуктів. Показано зв'язки базового ядра SWEBOOK, RMBOOK і стандартів, а також їхнє призначення в інфраструктурі менеджменту проекту.

Підручник призначається для навчання предмету «Програмна інженерія» у вищих навчальних закладах України. Його мета — навчити теорії і практиці створення комп'ютерних програм у сучасних інструментальних середовищах. Автор посібника має надію, що підручник буде не тільки допомагати виконувати практичні програмні розробки, які дуже часто є рутинними роботами з програмування щодо певного проекту, а й сприяти розвитку творчих здібностей студентів і бажанню створювати нові теорії в комп'ютерних науках.

Наведемо окремі нові задачі, які є перспективними на наступні десятиріччя:

- 15 річний міжнародний проект — теорія і практика верифікації усіх видів продуктів та їхнє накопичення у Інтернет-бібліотеках [11, 12];
- узагальнення технологічних засобів побудови проектів передових міжнародних фірм для підняття теоретичного подання процесу виробництва в них;
- розроблення теоретичного і прикладного підґрунтя мовно-орієнтованому програмуванню (специфікації специфіки доменів);
- комп'ютеризація математичних, логіко-алгебраїчних та обчислювальних знань тощо.

Зміст підручника обговорювався на всеукраїнської конференції викладачів і студентів «Програмна інженерія — напрями навчання», проведеної Національним авіаційним університетом (3–5 грудня 2007 р.), а також оприлюднений на семінарі викладачів і студентів (10 грудня 2007 р.) та на лекціях спецкурсу факультету кібернетики у Київському Національному університеті імені Тараса Шевченка.

Напрямки навчання. Базові положення щодо навчання Програмної інженерії подано в міжнародній

програмі Computing Curricula 2001, 2004 [13]. У ній містяться програми навчання як спеціальності, починаючи з першого курсу, так і як предмету загального навчання програмної інженерії на факультеті інформатики. Автор має досвід навчання програмної інженерії за програмою SE201 Curricula 2001. Цю програму рекомендовано як типовий факультативний навчальний план з програмної інженерії. Вона включає 12 тем:

1. Проектування ПС.
2. Інтерфейси застосувань.
3. Програмні засоби й оточення.
4. Процеси розробки ПС
5. Вимоги до ПС.
6. Перевірка відповідності ПС.
7. Методи еволюції ПС.
8. Керування програмними проектами.
9. Компонентно-орієнтована розробка.
10. Формальні методи.
11. Надійність і якість ПС.
12. Підходи до розробки спеціалізованих систем (не обов'язкова).

Усі наведені теми, окрім 12-ої, представлено в розглянутому підручнику і вони можуть використовуватися для практичної підготовки курсу лекцій для студентів.

Автор висловлює щире подяку конкурсній комісії «Вчитель-новатор» 2007 року під егідою компанії «Microsoft Україна», яка оцінила підручник у номінації викладачів вищих навчальних закладів (третє місце) і надала можливість надрукувати статтю про зміст нової наукової та інженерної дисципліни й підходу до її навчання у цьому журналі.

Література

1. Jacobson I. Object-oriented Software Engineering. A use case Driven Approach, Revised Printing. — New York: Addison Wesley, Publ. Co. — 1994. — 529 с.
2. Андон Ф.И., Лаврищева Е.М. Методы инженерии распределенных компьютерных приложений. — К.: Наукова думка, 1998. — 228 с.
3. Бабенко Л.П., Лаврищева К.М. Основы программной инженерии. Посібник. — К.: Знання, 2001. — 269 с.
4. Соммервил И. Инженерия программного обеспечения. — Москва+Санкт-Петербург+Киев: Изд. дом «Вильямс». — 2002. — 623 с.
5. Грищенко В.Н., Лаврищева Е.М. Методы и средства компонентного программирования//Кибернетика и системный анализ. — 2003. — №1. — С. 39–55.
6. Лаврищева Е.М. Методы программирования. Теория, инженерия, практика. — К.: Наукова думка, 2006. — 450 с.
7. Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Суслев // 2-е изд. — К.: Академперіодика. — 2007. — 672 с.
8. Лаврищева Е.М., Коваль Г.И., Коротун Т.М. Подход к управлению качеством программных систем обработки данных // Кибернетика и системный анализ. — 2006. — №5. — с. 174–185.
9. Задорожна Н.Т., Лаврищева К.М. Менеджмент документообігу в інформаційних системах освіти. — К.: Педагогічна думка, 2007. — 220 с.
10. Лаврищева К.М. Програмна інженерія. — К.: BHV, 2007. — 413 с.
11. Вудкок Д. Первые шаги к решению проблемы верификации программ // Открытые системы. — 2006. — №8. — С. 36–43.
12. Hoare T., Misra J. Verified software: Theories, Tools, Experiments. Vision of Grant Challenge project. — Microsoft Research Ltd and the University of Texas at Austin, 2005. — 1–43 с.
13. Рекомендации по преподаванию программной инженерии и информатики в университетах. — Computing Curricula-2001: Computer Science. — Перев. с англ. — Интернет— Университет Информационных технологий. — М.: 2007. — 462 с.