

УДК 378.14:372

І. О. Бардус,

кандидат педагогічних наук, доцент, докторант
(Українська інженерно-педагогічна академія)
irina.bardus@gmail.com

**ВИЗНАЧЕННЯ ЗАКОНОМІРНОСТЕЙ РОЗВИТКУ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ ТЕХНІКИ ЯК УМОВА
ФУНДАМЕНТАЛІЗАЦІЇ ПРОФЕСІЙНОЇ ПІДГОТОВКИ МАЙБУТНІХ
ФАХІВЦІВ У ГАЛУЗІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Анотація

Професійна підготовка майбутніх ІТ-фахівців до продуктивної професійної діяльності потребує фундаменталізації комп'ютерних дисциплін. У статті систематизовано та класифіковано за призначенням програмне забезпечення комп'ютерної техніки. Доведено необхідність ретроспективного аналізу кожного виду програмного забезпечення для фундаменталізації професійної підготовки майбутніх фахівців у галузі інформаційних технологій. На основі розроблених критеріїв визначено загальні закономірності розвитку програмного забезпечення комп'ютерної техніки.

Ключові слова: програмне забезпечення, технічна система, критерії розвитку, фундаменталізація, ІТ-фахівець.

Summary

Professional training of future IT professionals for productive professional activities requires the fundamentalization of computer disciplines. The article systematized and classified computer software software according to its purpose. The necessity of retrospective analysis of each kind of software for the fundamentalization of professional training of future specialists in the field of information technologies is proved. On the basis of the developed criteria, general laws governing the evolution of software for computer equipment are defined.

Key words: software, technical system, criteria of development, fundamentalization, IT specialist.

Постановка проблеми. Професійна діяльність майбутніх фахівців у галузі інформаційних технологій (ІТ) пов'язана з розробкою нового або використанням вже існуючого програмного забезпечення (ПЗ) комп'ютерної техніки. Але, враховуючи те, що програмне забезпечення постійно оновлюється, на момент закінчення студентом університету велика частина набутих ним знань встигає застаріти. Через це випускникам вищих навчальних закладів для того, щоб конкурувати на ринку праці із досвідченими інженерами-програмістами, необхідно самостійно оволодівати новими для них технологіями розробки програмного забезпечення, що вдається далеко не кожному. Розв'язати цю проблему можна лише за умови фундаменталізації професійної підготовки майбутніх ІТ-фахівців на основі універсальних загальнометодологічних знань і вмінь. Однією з умов фундаменталізації освіти, на нашу думку, є визначення закономірностей розвитку об'єктів професійної діяльності та технологій їх створення.

Аналіз останніх досліджень і публікацій. Концептуальними для аналізу розвитку програмного забезпечення є дослідження таких фахівців

з системного аналізу та технічних систем, як Г. Альтшуллера, А. Голубенка, Я. Дітріха, В. Зайончика, М. Лазарева, К. Сороки та ін. Вченими емпіричним шляхом визначено критерії та закони розвитку матеріальних технічних систем, однак ці закони не можуть бути використані для опису розвитку програмного забезпечення, оскільки воно є ідеальним об'єктом. **Мета** статті – систематизувати та класифікувати існуюче програмне забезпечення комп'ютерної техніки, визначити критерії і закономірності його розвитку на основі ретроспективного аналізу.

Виклад основного матеріалу. Оскільки комп'ютер складається з окремих частин ("елементів"), які виконують свою функцію (введення, виведення, зберігання, обробка інформації) і з'єднані між собою за допомогою системного інтерфейсу у вигляді чипсету (інформаційний зв'язок) та системи шин (електричний зв'язок), то його можна вважати "технічною системою" [3; 5; 9]. Фізичні компоненти комп'ютера прийнято називати апаратним забезпеченням (hardware), а команди, які керують цими компонентами, програмним забезпеченням (software) [2].

Програмне забезпечення можна визначити як групу взаємопов'язаних та взаємодіючих програм, які призначені для вирішення конкретних завдань у конкретній предметній області [1; 2; 7].

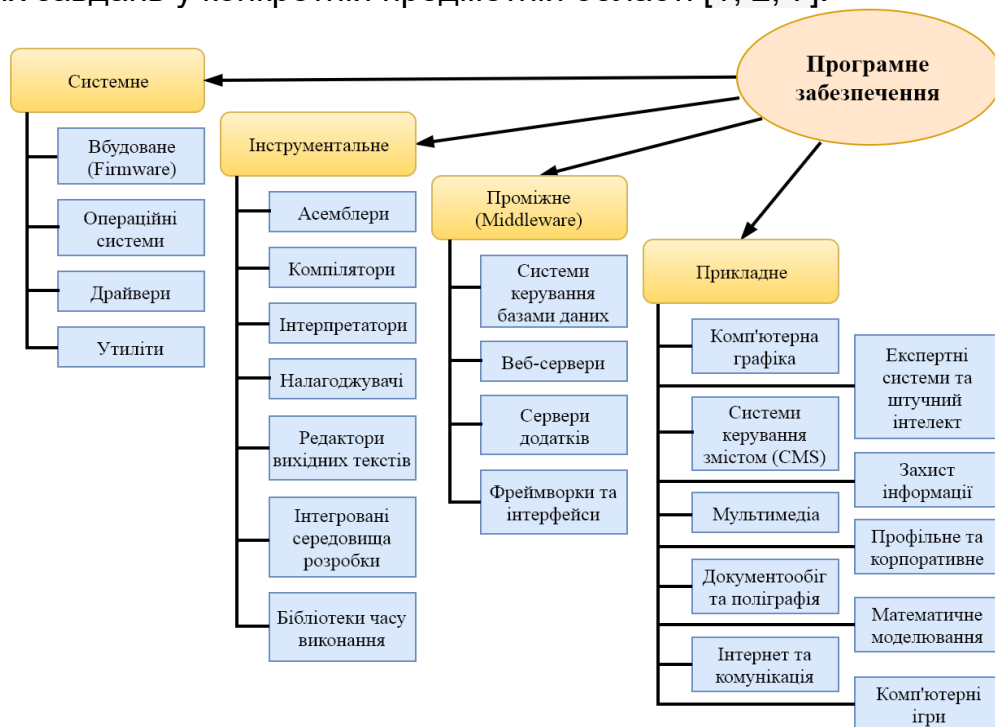


Рис. 1. Класифікація програмного забезпечення комп'ютерної техніки

Усе програмне забезпечення залежно від його призначення можна поділити на чотири категорії: системне [1; 2; 7], інструментальне [12], проміжне [11] та прикладне [1; 2; 7] (рис. 1).

Системне ПЗ забезпечує управління компонентами комп'ютерної системи такими, як процесор, оперативна пам'ять, пристрої введення-виведення інформації, мережеве обладнання, виступаючи "міжшаровим інтерфейсом", з одного боку якого апаратура, а з іншого – додатки

користувача [2; 7].

Інструментальне програмне забезпечення призначене для проектування, розробки і супроводу програм [12].

Проміжне ПЗ (Middleware) надає послуги для програмних додатків, крім тих, які доступні з операційної системи. Його можна охарактеризувати як “програмний клей”, що спрощує розробникам програмного забезпечення реалізацію комунікацій і введення / виведення інформації [11].

Прикладне програмне забезпечення призначене для виконання певних користувацьких завдань і розраховане на безпосередню взаємодію з користувачем [1; 2; 7].

Комп’ютерні програми являють собою набір інструкцій для зчитування апаратними пристроями комп’ютера, які приводять його у дію для досягнення певної мети або результату [1; 2; 7].

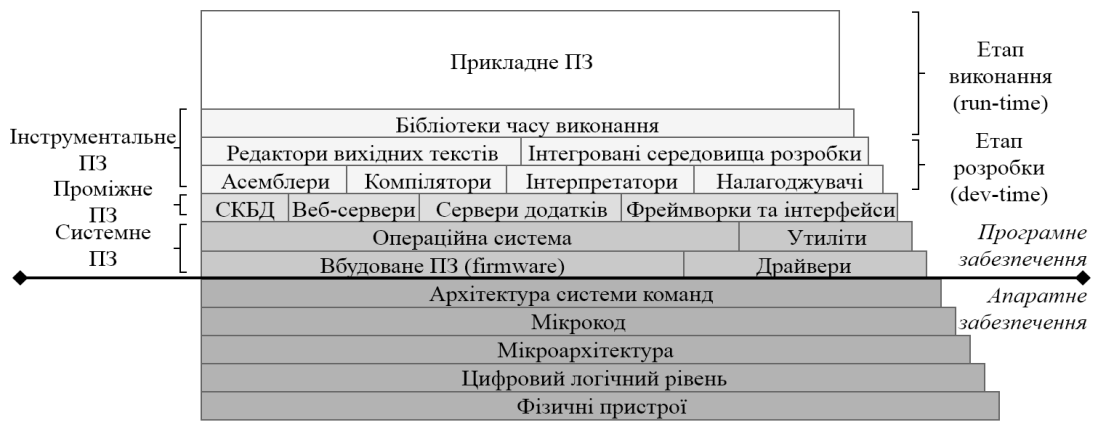


Рис. 2. Ієрархічна структура програмно-апаратного забезпечення комп’ютерної техніки

Проте не кожна комп’ютерна програма безпосередньо взаємодіє з апаратною частиною комп’ютера. На рис. 2. представлено ієрархічну структуру програмно-апаратного забезпечення комп’ютерної техніки. Кожен рівень ієрархії (шар) містить певне апаратне або програмне забезпечення комп’ютера. Кожен наступний шар програмного забезпечення може взаємодіяти тільки із двома сусідніми (нижнім або верхнім).

Майбутні IT-фахівці, особливо з програмної інженерії, повинні добре знати функціонал і розуміти особливості взаємодії системного, інструментального, сполучного та прикладного програмного забезпечення з апаратними компонентами комп’ютера.

Так, наприклад, розробнику пакетів тривимірної обробки зображень, окрім інструментального програмного забезпечення, необхідно використовувати різноманітні пакети прикладних програм і бібліотек, зокрема, бібліотеки моделювання фізичних процесів (наприклад, Navos), бібліотеки оптико-геометричного розрахунку освітлення сцени (ray-tracing). Також для функціонування графічних програм використовується проміжне програмне забезпечення, в якості якого виступають апаратно-програмні бібліотеки DirectX або OpenGL, що забезпечують високорівневі

абстрактні інтерфейси до апаратного рівня графічної підсистеми. Крім того, ефективний доступ програмних компонентів графічних пакетів до апаратних ресурсів комп'ютера забезпечують системні програми (зокрема, операційна система).

Для організації ефективної роботи графічних пакетів розробнику та користувачу необхідно знати, як залежить продуктивність і швидкість виконання певних задач від ресурсів комп'ютера. Наприклад, комп'ютерне графічне моделювання та анімація, що передбачає моделювання сцен з тривимірними об'єктами, текстурування об'єктів та їх оточення, створення шейдерів, обчислення освітлення всіх елементів сцени, ключову та скелетну анімацію, розрахунок одягу, волосся та інших ефектів, заснованих на системі частинок; розрахунок фізичної взаємодії об'єктів сцени між собою; загальний рендеринг анімаційної сцени та її подальшу обробку потребує значних ресурсів комп'ютерної техніки. Так, від кількості та продуктивності графічних карт залежить швидкість прорахунку всіх етапів створення сцени, а особливо етапу рендерингу. Чим більший обсяг оперативної пам'яті комп'ютера, тим більший обсяг об'єктів сцени доступний для маніпуляцій без додаткового застосування віртуальної пам'яті. Оперативна пам'ять також використовується для кешування даних під час прорахунку сцени на етапі рендерингу. Швидкість обробки сцени на всіх етапах на пряму залежить від продуктивності центральних процесорів та кількості ядер у кожному із них. Також загальна швидкість обробки сцени залежить від швидкості зчитування та запису пристроїв довготривалого зберігання інформації (HDD та SSD). Отже, майбутньому інженеру-програмісту необхідно оволодіти великою кількістю знань з функціонування, технологій розробки, взаємодії кожного виду програмного забезпечення між собою та фізичними частинами комп'ютерної техніки.

На нашу думку, для ефективної професійної підготовки майбутніх ІТ-фахівців до продуктивної професійної діяльності необхідно розробити єдиний системний підхід до представлення навчального матеріалу різних комп'ютерних дисциплін, який дозволить сформувати в студентів фундаментальні методологічні знання та вміння зі створення нового програмного забезпечення.

Для системного опису понять комп'ютерної дисципліни, на нашу думку, доцільно застосувати розроблену Я. Дитріхом [4], і вдосконалену М. Лазаревим [6] універсальну ієрархічну модель технічного об'єкта $P = \{R, S, D, H\}$ на основі семантичних ознак (призначення (R), склад (S), принцип дії (D) та характеристики (H)). У випадку, коли під технічним об'єктом виступає програмне забезпечення, його семантичними ознаками є: призначення (R), внутрішня архітектура та сукупність зовнішніх бібліотек і компонентів, на які спирається ПЗ (S), алгоритми, математичні теорії та закони, на основі яких побудована програма (D), та характеристики (H).

Науковцями [5; 10] доведено, що важливу роль у підвищенні ефективності професійної діяльності фахівця та її результатів при пошуку нових технічних рішень відіграють знання закономірностей розвитку технічних систем, вміння їх аналізувати і використовувати для виявлення резервів їх розвитку, визначення доцільності вдосконалення або створення принципово нових технічних систем. Оскільки програмне

забезпечення є підсистемою технічної системи “комп’ютер”, тому його вдосконалення відбувається за принципом прогресивного розвитку технічних систем, а саме: для кожного нового покоління систем поліпшуються визначені показники оцінки критеріїв (Н) їх розвитку за умови не погіршення, якщо це можливо, інших показників.

Сучасні складні технічні системи характеризуються великою кількістю показників для оцінки ефективності їхнього функціонування. Виділяють чотири основні групи критеріїв розвитку технічних систем [5]:

- функціональні критерії, що характеризують найважливіші показники технічних систем (продуктивність, точність, надійність, спеціальні);

- технологічні критерії, що визначаються можливостями (ступенем складності) проектування і виготовлення технічних систем (трудомісткість виготовлення, технологічні можливості, використання матеріалів, декомпозиція);

- економічні критерії, що визначають економічну доцільність проектування й експлуатація технічних систем (витрати матеріалів, витрати енергії, витрати на підготовку й одержання інформації, габаритні розміри);

- антропологічні критерії, характерні для систем “людина-машина” (ергономічність, безпека, екологічність).

Невідповідність технічної системи певним параметрам перелічених показників (Н) викликає її розвиток за рахунок зміни її структури (S) або принципу дії (D).

Визначення закономірностей розвитку програмного забезпечення доцільно проводити на основі порівняння кількісних та якісних показників функціональних, технологічних, економічних та антропологічних критеріїв. Проте розроблені в дослідженнях [5; 10] критерії оцінки технічної системи (продуктивність, точність, надійність, трудомісткість виготовлення, технологічні можливості, використання матеріалів, декомпозиція, витрати матеріалів, витрати енергії, витрати на підготовку й одержання інформації, габаритні розміри, ергономічність, безпека, екологічність) є коректними тільки для матеріальних систем.

Такі критерії, як використання матеріалів, витрати матеріалів, витрати енергії, габаритні розміри, екологічність для розгляду ідеальних технічних систем (програмного забезпечення) не можна застосовувати, оскільки вони вказують на характеристики матеріальних об’єктів. Інші ж критерії (продуктивність, точність, надійність, трудомісткість виготовлення, технологічні можливості, декомпозиція, витрати на підготовку й одержання інформації, ергономічність, безпека), на нашу думку, можуть бути використані для опису ідеальних технічних систем тільки після їх адаптації до відповідної предметної галузі. Тому наступним кроком для визначення закономірностей розвитку програмного забезпечення комп’ютерної техніки розробимо систему критеріїв їх ефективності.

Програмне забезпечення являє собою продукт діяльності людини, спрямований на розв’язання певної задачі. Тому критерії ефективності розробленого програмного продукту мають описувати технологію створення (технологічні критерії), умови функціонування (функціональні

критерії), економічні витрати на його розробку (економічні критерії) та показники зручності й ефективності його використання користувачем.

До технологічних критеріїв, що описують можливості (ступінь складності) проектування і виготовлення програмних систем, на нашу думку, належать: об'єм коду, складність коду, складність процесу розробки, час розробки, відкритість коду, ступінь декомпозиції.

До функціональних критеріїв, що визначають умови забезпечення ефективного функціонування програмного забезпечення, віднесемо: загальні (продуктивність, швидкість виконання, ресурсоемність, ступінь інтегрованості компонентів, ефективність використання (кількість операцій для досягнення результату), спеціальні (надійність, розподіленість обробки, розподіленість даних, багатозадачність, універсальність коду, сумісність програмна, сумісність інтерфейсів, ступінь структурованості даних, ресурсозабезпеченість, інтегрованість даних, багатофункціональність).

До економічних критеріїв, які визначають економічну доцільність проектування й експлуатації програмних систем, належать: вартість розробки ПЗ, вартість ПЗ.

До антропологічних критеріїв, що визначають ефективність та зручність користування, психофізіологічну сумісність інтерфейсу ПЗ, доцільно віднести: ергономічність, складність навчання, інформованість про стан системи, однаковість та стандартизованість, запобігання помилок, довідникові матеріали. Крім того, оскільки програмне та апаратне забезпечення є діалектичними компонентами комп'ютера і за філософським законом єдності та боротьби протилежностей розвиток однієї з них впливає (або викликає) розвиток іншої, тому при ретроспективному аналізі програмного забезпечення серед причин (технологічних, функціональних, економічних, антропологічних) його еволюції також необхідно зазначати і вдосконалення в галузі апаратних пристроїв комп'ютерної техніки.

Ретроспективний аналіз програмного забезпечення має будуватися на основі причин виникнення кожного покоління (показників, які необхідно було покращити) із зазначенням нової структури, принципу дії (математичних алгоритмів, законів та теорій) або призначення. Порівняння динаміки показника протягом усього часу еволюції дозволить визначити закономірності розвитку програмного забезпечення.

Приклад ретроспективного аналізу баз даних та систем керування базами даних наведено на рис. 3.

Отже, з ретроспективного аналізу баз даних та систем їх керування можна виділити такі закономірності їх розвитку:

Поступове ускладнення структурних моделей, зумовлене низьким рівнем структурованості даних, тобто необхідністю більш повного відображення предметної області у відповідних моделях даних.

Тенденція до розподіленого зберігання та обробки даних зумовлена необхідністю обробки дуже великих обсягів даних, що неможливо здійснити на одному комп'ютері.

Поява систем керування базами даних з різним рівнем цільового масштабу даних, пов'язана з необхідністю зберігання та обробки даних різного масштабу (обсягу та складністю задач).

Еволюція баз даних та СКБД

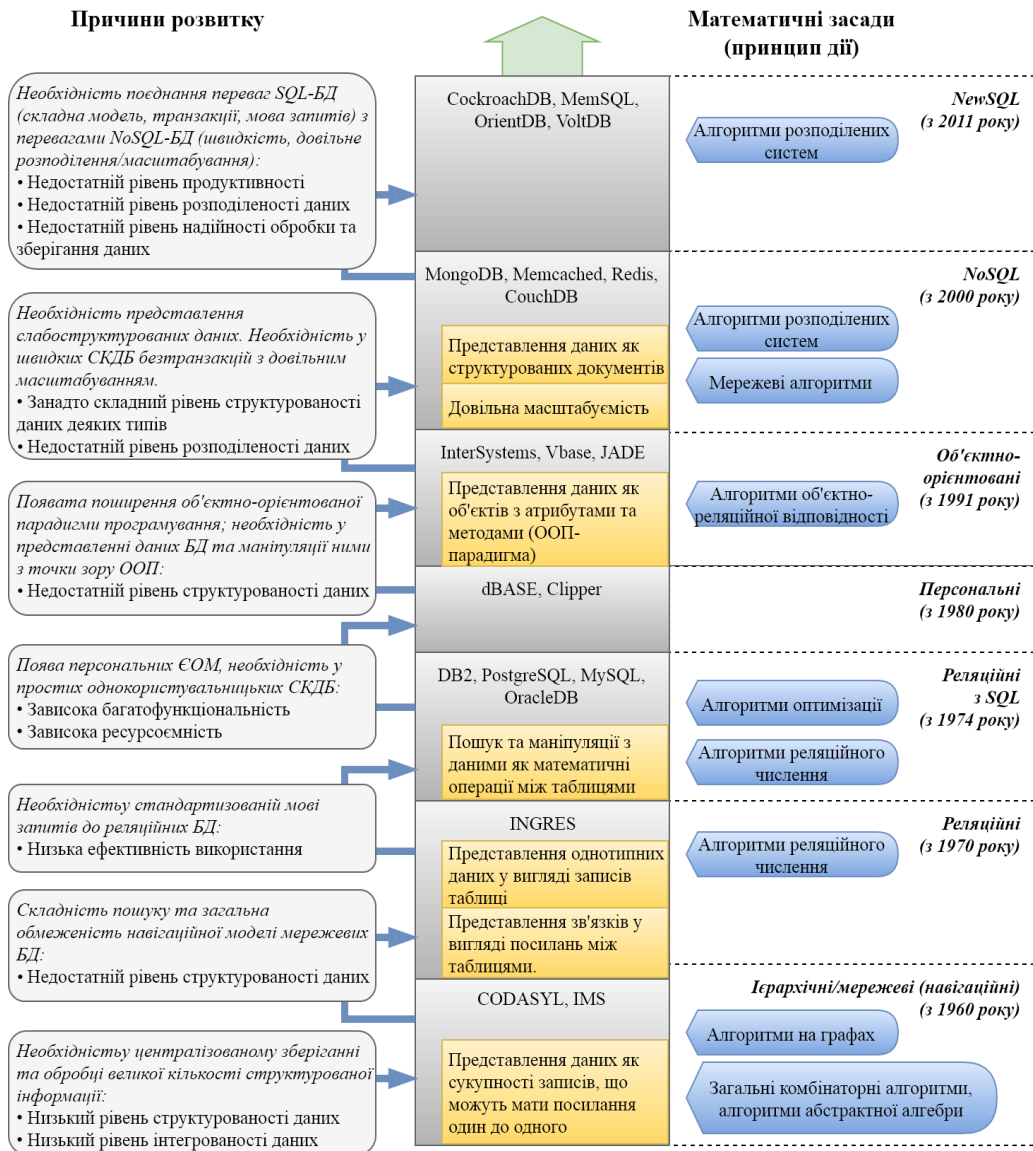


Рис. 3. Ретроспективний аналіз баз даних та систем

Поява різноманітних моделей баз даних, заснованих на різноманітних принципах представлення та обробки баз даних, зумовлена різноманіттям предметних областей та типів поставлених перед системою керування базами даних задач.

Якщо провести ретроспективний аналіз усього програмного забезпечення за наведеними вище критеріями, то можна визначити загальні закономірності розвитку програмного забезпечення, а саме:

- збільшення обсягів даних, що обробляються програмним забезпеченням;
- збільшення функціональності по мірі розвитку продукту;
- ускладнення коду через поступове збільшення функціональних можливостей програмного продукту, а також підтримки нових парадигм та технологій програмної інженерії;
- збільшення ресурсоемності ПЗ через розширення функціональних можливостей та збільшення обсягів даних, що обробляються;

□ збільшення часу виконання програм (при порівнянні старого та нового програмного забезпечення на одній апаратній платформі) за рахунок збільшення даних, що потребують обробки, росту обсягів програмного коду та вимог до апаратної частини, що зростають.

Висновки. Підвищення якості професійної підготовки майбутніх фахівців у галузі інформаційних технологій до продуктивної професійної діяльності потребує фундаменталізації комп'ютерних дисциплін на основі загальнометодологічних знань та вмінь. Однією з умов фундаменталізації професійної підготовки ІТ-фахівців є визначення закономірностей розвитку програмного забезпечення.

Перспективи подальших пошуків у напрямку дослідження. Ретроспективний аналіз та закономірності розвитку програмного забезпечення будуть нами використані при розробленні математичних засад фундаменталізації професійної підготовки майбутніх ІТ-фахівців.

ЛІТЕРАТУРА

1. Батоврин В. К. Толковый словарь по системной и программной инженерии / В. К. Батоврин. – М. : ДМК Пресс, 2012. – С. 280.
2. Васильев В.Г. Введение в системное программное обеспечение: учеб. пособие / В. Г. Васильев. – Тверь : ТГТУ, 2009. – 160 с.
3. Голубенко А. Л. Теория технических систем: учеб. пособие для вузов / А. Л. Голубенко, А. С. Петров, А. Л. Кашура. – К.: Аристей, 2004. – 239 с.
4. Дитрих Я. Проектирование и конструирование: системный подход / Я. Дитрих. – М.: Мир, 1981. – 456 с.
5. Заёнчик В.М. Основы творческо-конструкторской деятельности: Методы и организация : учебник для студ. высш. учеб. заведений / В. М. Заёнчик, А. А. Карачёв, В. Е. Шмелёв. – М. : Издательский центр “Академия”, 2004. – 256 с.
6. Лазарев М. І. Полісистемне моделювання змісту технологій навчання загальноінженерних дисциплін : монографія / М. І. Лазарев. – Х. : Вид-во НФаУ, 2003. – 356 с.
7. Липаев В. В. Проектирование программных средств : учеб. пособ. / В. В. Липаев. – М.: Высшая школа, 1990. – 302 с.
8. Сорока К. О. Основы теории систем і системного аналізу : навч. посіб. / К.О. Сорока. – Х.: ХНАМГ, 2004. – 291 с.
9. Хубка В. Теория технических систем / В. Хубка. – М. : Мир, 1987. – 202 с.
10. Шанс на приключение / Сост. А. Б. Селюцкий. – Петрозаводск : Карелия, 1991. – 304 с.: ил. – (Техника – молодежь – творчество).
11. “What is Middleware?” [Електронний ресурс]. – Режим доступу : web.archive.org/web/20120629211518/http://www.middleware.org/whatis.html
12. Workshop on Software Development Tools for Petascale Computing, 1-2 August 2007, Washington, DC [Електронний ресурс]. – Режим доступу : <http://www.csm.ornl.gov/workshops/Petascale07/>

Стаття надійшла до редакції 26.09.2017