

УДК 004.8 : 007 : 519.681

Степура Ігор Володимирович, старший лаборант Інституту психології імені Г.С. Костюка НАПН України

e-mail: istep@ukr.net

ОБ'ЄКТНО-ОРІЄНТОВАНИЙ ПІДХІД: ІСТОРІЯ ТА ДЕЯКІ ДОДАТКИ В ПСИХОЛОГО-ПЕДАГОГІЧНИХ ГАЛУЗЯХ

Анотація

У статті розповідається про об'єктно-орієнтовану парадигму інтеграції даних (в формах системного аналізу, програмування, проектування), яка знаходить своє застосування в різних галузях науки і техніки. Об'єктно-орієнтована парадигма склалася на перетині математики, логіки, кібернетики і психології в 60-ті рр. ХХ століття. Історично важливий внесок в об'єктно-орієнтовану парадигму наступних дослідників-періопрохідців: А.Сазерленд, К. Ногорд, О. Даль, А.Кей, Д.Інглас, Т.Каглер, А.Голдберг. Вони запропонували концепції двох основоположних мов програмування: Simula 67 і Smalltalk, концепції яких стали джерелом багатьох сучасних мов. Об'єктно-орієнтований підхід до організації програми дозволяє групувати певні фрагменти інформації разом з часто використовуваними функціями або діями, пов'язаними з цією інформацією. Він вказує загальний принцип об'єкта в ООП, як особливого об'єктного типу даних. З ростом популярності мов програмування, де застосовується технологія об'єктно-орієнтованого програмування, перед педагогами постало завдання навчання студентів до цього підходу, але пізніше стали замислюватися над застосуванням підходу як дидактичного, як теоретичної основи для розробки методології і технології ефективного структурування змісту навчання. На пострадянському просторі ведуться роботи по прикладенню підходу для потреб навчання (О. Баранова, І.Аржанов, О. Степанов). Є застосування в галузі організаційної психології (В.Дорохов, В. Копнов, О. Валєєва).

Ключові слова: об'єктно-орієнтована парадигма, система, інформатика, об'єкт, А.Сазерленд, об'єктно-орієнтований підхід до процесу навчання, мова програмування.

Степура Игорь Владимирович, старший лаборант Института психологии имени Г.С. Костюка НАПН Украины

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОХОД: ИСТОРИЯ И НЕКОТОРЫЕ ПРИЛОЖЕНИЯ В ПСИХОЛОГО-ПЕДАГОГИЧЕСКИХ ОБЛАСТЯХ

Аннотация

В статье рассказывается об объектно-ориентированной парадигме интеграции данных (в формах системного анализа, программирования, проектирования), которая находит своё применение в различных отраслях науки и техники. Объектно-ориентированная парадигма сложилась на пересечении математики, логики, кибернетики и психологии в 60-е гг. XX века. Исторически важен вклад в объектно-ориентированную парадигму следующих исследователей-первопроходцев: А.Сазерленд, К. Ногорд, О. Даль, А.Кей, Д.Инглас, Т.Каглер, А.Голдберг. Они предложили концепции двух основополагающих языков программирования: Simula 67 и Smalltalk, концепции которых стали истоком многих современных языков. Объектно-ориентированный подход к организации программы позволяет группировать определенные фрагменты информации вместе с часто используемыми функциями или действиями, связанными с этой информацией. Он указывает общий принцип объекта в ООП, как особого объектного типа данных. С ростом популярности языков программирования, где применяется технология объектно-ориентированного программирования, перед педагогами встала задача обучения студентов этому подходу, но позже стали задумываться над применением подхода как дидактического, как теоретической основы для разработки методологии и технологии эффективного структурирования содержания обучения. На постсоветском

пространстве ведутся работы в области применения подхода для нужд обучения (Е.В. Баранова, И.Н.Аржанов, А.Г. Степанов). Есть применения в области организационной психологии (В.Н.Дорохов, В.А. Копнов, Е.В. Валеева).

Ключевые слова: *объектно-ориентированная парадигма, система, информатика, объект, А.Сазерленд, объектно-ориентированный подход к процессу обучения, язык программирования.*

Stepura Igor Volodymyrovych, senior laboratory assistant, Cognitive Psychology department at the Institute of Psychology named by G.S. Kostyuk of NAPS of Ukraine

THE OBJECT-ORIENTED APPROACH: HISTORY AND SOME APPLICATION IN PSYCHOLOGY AND EDUCATION

Summary

The article describes the object-oriented paradigm of data integration (in the forms of systems analysis, programming, design), its application can be found in various fields of science and technology. The object-oriented paradigm is formed at the intersection of mathematics, logic, cybernetics and psychology in the 60th of the XX century. The next early researches made historically important contribution to the object-oriented paradigm: I.Sutherland, K. Nygaard, O. Dahl, A. Kay, D.Inglas, T.Kagler, A.Goldberg. They offered two fundamental concepts of programming languages: Simula 67 and Smalltalk, the concept of which were the source of many modern languages. The object-oriented approach to programming allows a programmer to group particular pieces of information together with frequently used functions or actions associated with that information. The general principle in the object-oriented approach is that an object is described as a special object data type. With increasing of popularity of the programming languages that use the object-oriented programming technology, teachers faced with the task to teach students to this approach, and later ideas appeared to use such approach as didactic one, as a theoretical basis for development of methodologies and

technologies for efficient structuring of learning content. In the ex-USSR countries, works are underway aimed at application of such approach for the needs of education (E.V. Baranova, I.N.Arzhanov, A.G. Stepanov). There are some applications in the field of organizational psychology (V.N.Dorohov, V.A. Kopnov, E.V.Valeeva).

Keywords: *object-oriented paradigm, system, information science, object, I.Sutherland, object-oriented approach in education, programming language.*

Постановка проблеми. Сучасний етап розвитку науки характеризується впровадженням в різні предметні галузі широких парадигмальних моделей, які потребують на практиці застосування певної методології інтеграції даних. Однією з основ для такої інтеграції послугувала концепція об'єктно-орієнтованого проектування (а також програмування (ООП) і аналізу складних систем) [11, с.196–217]. Об'єкт складається з структур даних (поля; знання) і сукупностей процедур, що їх обробляють (методи; дії). Об'єкти маніпулюють структурами даних, які містяться в них, і обмінюються інформацією між собою. Згодом ООП став застосовуватися в різних галузях: фізиці, математиці, системах управління, біології [4;5].

Аналіз літератури. Об'єктній парадигмі більш ніж півстоліття – перерахувати всі дослідження важко. Згадаймо тут лише основоположні роботи А.Кея, Г. Буча, Д. Рамбо, П. Коад [4;13;14]. Є і застосування в галузі гуманітарних наук, зокрема психолого-педагогічного спрямування. Відзначимо роботи О.В. Баранової, І. Аржанова, О. Степанова [1;2;3;10]. Є роботи в галузі організаційної психології (В.Дорохов) [6].

Метою нашого дослідження є вивчення історії появи ООП та його деякі прикладення в психолого-педагогічних дослідженнях.

Основний виклад. Об'єктно-орієнтована парадигма склалася на перетині математики, логіки, кібернетики та психології. До початку 60-х рр. XX століття було сформульовано дві концепції програмування: імперативна (в основі лежить поняття алгоритму та його виконання, це найбільш масова й

усім знайома модель) і функціональна (в центрі обчислення функцій в її математичному сенсі, алгебраїко-логічна методологія). При імперативному підході в пам'яті комп'ютера ми маємо змінні та стани, що призводять до різних результатів при виконанні алгоритму. Процес обчислення в функціональному програмуванні – це обчислення значень математичних функцій. Функції задаються у вигляді списку, він і направляє весь обчислювальний процес. Маючи функцію, її аргумент і опис функції завжди отримуємо однозначний результат. Замість покрокового алгоритму розв'язання задачі, задається специфікація рішення задачі (опис, проблема, результат). Якщо імперативний підхід – живе втілення кібернетичної концепції автоматів, то функціональний – теорії символічних перетворень. Функціональне програмування входить як частина до складу «декларативного програмування». Формально кажучи, декларативна програма є теорема, яку треба довести машинними методами. Автором функціонального програмування вважається Джон Маккарті (1927–2011), він же і запропонував термін «штучний інтелект». У 1958 р. була створена функціональна мова програмування Lisp. Lisp працює з атомами (символами й числами), що об'єднані в списки. Списки організовані так, щоб зручно було зберігати ієрархічні (деревовидні) структури. Символ в Lisp це сукупність «слотів» у вигляді імен, функціональних слотів – так називають пов'язані з символами функції, й значення (дані). Lisp-програма складається з послідовності виразів (форм). Lisp дозволяє створювати програми, які модифікують інші програми, тому мова використовувалась для створення інтелектуальних систем. Логічна основа Lisp: лямбда-числення А.Чорча [11].

В 1963 р. в США була створена перша система автоматизованого проектування (САПР, CAD) за допомогою якої можна було створювати тривимірні моделі та оперувати ними. Називалася система «Sketchpad». Її автор – Айван Сазерленд. «Sketchpad» – була перша програма в світі з графічним інтерфейсом. Ґрунтуючись на методі візуального проектування Сазерленд ввів поняття «об'єкт» як графічного примітиву до елементів якого

(ліній, дуг, кіл, кривих, ламаних) можна було прив'язати ті чи інші математичні функції, проводячи з ними розрахунки. Об'єкт приймався за певну цілісність; при тому можна було будувати дублікати таких об'єктів. У разі ланцюжка дублікатів змінюючи що-небудь у вихідному об'єкті можна було змінити цю властивість і в усьому подальшому ланцюжку («успадкування»). Через вікна на екрані можна було оперативно змінювати розміри графічної моделі – тобто мав місце перший інтерактивний діалог людини й ЕОМ. Зв'язок об'єкта та функції був близький до ідей функціонального програмування. Бо ж тривимірну модель можна представити як візуалізацію певної математичної функції. Сазерлендівський «об'єкт» – абстрактна модель у якій дані об'єднані з функцією, що їх обробляє. Протягом 60-х стала складатися концепція типу даних: константа, символ, ціле число, дійсне число, список тощо. Помітним явищем в світі комп'ютерів стає мова програмування Algol 60. У 1967 р. в Норвегії Крістен Ногорд і Оле-Йохан Даль створюють першу об'єктно-орієнтовану мову Simula 67 (як розширення стандарту Algol 60). В Simula 67 був тип даних «об'єкт». Це була новаторська мова, синтаксично схожа на сучасні, але широкого визнання вона тоді не отримує. У 1966–68 рр. А.Сазерленд співпрацює з університетом Юти (США). В його дослідницькій групі працює програміст, і у вільний час джаз-музикант, Алан Кей (р. 1940). Тема його досліджень – діалог користувача та комп'ютера через графічний інтерфейс. Сазерленд захоплює молодого науковця своєю ідеєю «об'єктів» та когнітивними структурами взагалі. Кей освоює мову Simula-67. У цей час він приходить до принципу «біологічної аналогії»: ідеальний комп'ютер має бути подібним до живого організму, де кожна клітина індивідуальна, але всі разом вони утворюють єдину систему, здатну до перекомпонування й зміни структури. У 1968 р. Кей знайомиться з Сеймуром Пейпертом. У 1959–1963 рр. С. Пейперт працював в Женевському університеті під керівництвом Ж. Пьяже де розробляв питання застосування математичних підходів до вивчення розвитку дитини. Після цього він їде в США до Массачусетського

університету, де разом з М. Мінскі засновує лабораторію штучного інтелекту. У 1967 р. (разом з І. Харель) Пейперт створює мову Лого для цілей розвитку мислення дітей і формування в них навичок програмування. Пізніше Пейперт широко відомий автор книг про розвиток і навчання дитини, із взаємодії дитини і обчислювальної техніки. Велика заслуга Пейперта в тому, що він вніс в роботи з об'єктного підходу психологічний струмінь. Саме С.Пейперт спонукав Кей до вивчення праць Ж.Пиаже, Дж. Бруннера, Л. Виготського.

А. Кей приступає до розробки ще однієї об'єктно-орієнтованої мови програмування, відомої нині як Smalltalk (разом з Д.Інглас, Т.Каглером, А.Голдберг). За планами Smalltalk (як і Лого) повинні були вивчати діти: Кей навіть поставив серії дослідів за участю школярів. Smalltalk повинен був стати мовою концептуального навчального мікрокомп'ютера для школяра і студента [14]. У 1970-х Кей працюючи в компанії Херох, висловив думку, що екран комп'ютера – найважливіше середовище поширення знань як і папір, він висунув ідеї ноутбука і планшета. Саме Кей сформулював також принцип віконного інтерфейсу, який ми всі сьогодні знаємо. З 1984 р. працював у фірмі Apple. Стандарт мови Smalltalk склався у 1980 р. та при цьому він відрізнявся від Simula 67 більш суттєвим радикалізмом порівняно з традиційними мовами програмування. Згідно Smalltalk всі елементи програми є об'єкти, які обмінюються повідомленнями (теж «об'єктами» за структурою). Всі змінні, оператори, абстракції, реалізовані як об'єкти, тому все середовище, засоби мови можуть бути модифіковані під задачу. У Smalltalk з'явилися «класи» – особливий абстрактний тип даних, свого роду модель, що описує базові властивості майбутніх «об'єктів». Структурно вони побудовані за тим же зразком формального «об'єкта» запропонованого Сазерлендом, але вони не беруть участь в ході виконання програми, а лише слугують для створення похідних конструкцій, які знову таки мають властивості «об'єкта» (є вислів: об'єкт – екземпляр класу) [4; 12, с.3-15].

Об'єктно-орієнтоване програмування (ООП) – це спосіб організації коду в програмі шляхом його угруповання у вигляді об'єктів, тобто окремих елементів, які включають інформацію (значення даних) й функцій визначених над ними. Об'єктно-орієнтований підхід до організації програми дозволяє групувати певні фрагменти інформації разом з часто використовуваними функціями або діями, пов'язаними з цією інформацією. Він вказує загальний принцип ООП-об'єкта як сутності, як особливого об'єктного типу даних. Це, по суті, формально-математичний конструкт, який не треба навантажувати сторонніми поняттями як-то «об'єктивність», «об'єкт фізичного світу» та ін. Приклади: ключ-відкриває, музика-програвач-відтворення і т.д. [4;9]

«Об'єкт як сутність» має властивість інкапсуляції – приховуванням від інших «об'єктів» і операцій в програмі своєї внутрішньої структури. Ми детально не знаємо пристрою замку, електричної схеми програвача. «Об'єкт як сутність» має свій проект, описаний через клас. Клас – це абстрактна модель ще не існуючої сутності (об'єкта). Вона також побудована на «об'єктних» принципах.

Для класів часто визначають окреме поняття інтерфейсу. Інтерфейс в загальному випадку це сукупність методів і правил взаємодії елементів системи. Інтерфейс визначає, як же будуть взаємодіяти між собою ці елементи. Знання інтерфейсу телефону: трубка, диск (кнопки), індикатор (іноді) дозволяє нам користуватися зв'язком в самих різних ситуаціях. У комп'ютерах ми стикаємося з визначенням графічного інтерфейсу, його знання допомагає користуватися функціями (функціоналом) тієї чи іншої програми. Розуміння стандартних основ інтерфейсів дає можливість швидко освоювати користувачам нові програми чи прилади. Інтерфейсом об'єкта називають набір операцій (методів), які він може здійснювати («поведінка») або які можна здійснювати над ним. Клас «число» має інтерфейс складений з арифметичних операцій; такий же інтерфейс мають похідні класи комплексне число, дійсне число й ціле число. Об'єкти, що мають однаковий інтерфейс

мають подібну поведінку. Інтерфейс можна визначити як ще один вид абстрактного класу. Це той же самий абстрактний клас, але який містить тільки абстрактні функції (методи). Ці функції задані узагальнено, тобто без конкретної реалізації. Вважається, що інтерфейс більш «абстрактний» ніж клас.

Д.Попов зазначає [9], що людина схильна думати інтерфейсами, а не класами (детальними проектами). Опис інтерфейсу «відкрити двері» переважає прагнення дізнатися як влаштований замок яким вона замкнена. Він згадує «класичний досвід з мавпою, яку привчили гасити вогонь водою з відерця; а потім поставили відерце на пліт посеред басейну, але мавпа все одно бігала по містку на пліт і черпала воду з відерця, замість того, щоб черпати воду прямо з басейну. Тобто мавпа використовувала клас Вода в відерці замість інтерфейсу «Вода » [9]. Отже інтерфейс – абстрактне поняття, конкуруюче з класом. Багато авторів вважають його навіть більш абстрактним ніж клас (проект, головний проект).

Інший важливий принцип успадкування – це властивість, що дозволяє описати новий клас на основі вже існуючого. Структура вихідного класу (базового, батьківського) переноситься на похідний клас, й утворюються послідовності класів або ж в загальному випадку ієрархія. Ієрархія класів – це структура багаточисельних пов'язаних класів, що визначає, які класи успадковують функції від інших класів. У разі ієрархії для довільного класу визначають верхній клас-прабатько, називаючи його надкласом, а нижній похідний – підкласом. Самий «нижній» клас підкласу не має. Часто вдається протягом усього процесу успадкування розглядати різні аспекти моделі й при цьому зберігати загальну незмінну структуру протягом усього ланцюга класів. Функціональність тут запозичується повністю. Але буває і випадок, коли набір функцій у всіх класах послідовності або дерева однаковий, але здійснюється лише часткове запозичення з їх набору. Простіше: успадкування дозволяє одному класу включати всі функції іншого класу (і, як правило, розширювати ці функції), або взяти лише частину. Успадкування

дозволяє модифікувати концепти моделі по вісі від загального до конкретного вже на рівні проекту.

Від проектної моделі (клас) потрібно переходити до конкретної реалізації. Побудовані на основі класу(ів) сутності (в практиці програмування сукупності виконуваного коду) об'єднуються з реальними даними й починають діяти, працювати. Отримані таким чином конструкції також називаються «об'єктами», хоча є і більш правильне визначення «екземпляр класу». Така синонімія часто заважає розрізнити конкретну реалізацію проекту від загального принципу «об'єкта як сутності» про який говорив Сазерленд. Ще одним принципом ООП є поліморфізм – здатність однієї і тієї ж функції до обробки даних різних типів. У загальному випадку маємо два типи поліморфізму. Перший – *ad hoc* поліморфізм (спеціальний), має на увазі виконання різного коду (частин програми) для кожного типу даних. Один вхід багато реалізацій (або один інтерфейс різний код) – ось його головний принцип. Параметричний поліморфізм – як правило, атрибут функціонального програмування. Тут виконується один і той же код для всіх можливих типів аргументів. Це досягається високим ступенем абстрактності даних і функцій. У ООП під поліморфізмом розуміють випадок використання об'єктів з однаковим інтерфейсом без інформації про їх типи та внутрішню структуру. Поліморфізм треба відрізняти від успадкування. При успадкуванні зміни вже закладені в проект, а при поліморфізм зміни відбуваються на рівні реалізації [4; 13].

Більш строго: поліморфізм припускає використання єдиного імені (ідентифікатора) при зверненні до об'єктів кількох різних класів, за умови, що всі вони є підкласами одного загального надкласу.

Об'єктна концепція отримала розвиток в логіці й класичній когнітивній психології. Перш за все «об'єкт як сутність» можна розглянути як параметричне розширення поняття фрейму. Фрейм тут отримує функціонально-динамічну складову. Знання стають дієвими. Логіки розглядали «відокремлений» об'єкта або ж об'єкти об'єднані в семантичні

мережі (Тей). Але сьогодні частіше їх все таки об'єднують в ієрархії. Ще на початку 1980-х рр. Х.Тиугу висував «об'єкт» – на роль моделі базового поняття. У роботі [12, с. 124–196] наводилася формалізація понять фізико-математичного циклу. Застосовують ООП і для безпосереднього аналізу складних систем і управління ними [8].

З ростом популярності мов програмування, де застосовується технологія об'єктно-орієнтованого програмування, перед педагогами постало завдання навчати цьому підходу студентів, але пізніше стали замислюватися над застосуванням ООП як дидактичної парадигми [1]. О.Г. Степанов за допомогою ООП розробив модель інформатики як предмета навчання [10]. У докторській дисертації О.В.Баранової [3] вже ставиться більш глобальне завдання: «Використання об'єктно-орієнтованого проектування як теоретичної основи для розробки методології і технології ефективного структурування змісту навчання засобами інформаційних технологій в умовах інтенсифікації інформаційних процесів» [3, с.3] і далі: «Аналіз можливих підходів на шляху вдосконалення навчання засобам інформаційних технологій привів до висновку про необхідність пошуку нового апарату, здатного забезпечити подання змісту навчання цієї предметної області відповідно до сучасних принципів дидактики і адекватно предметній області в умовах інтенсифікації інформаційних процесів» [3 , с.4]. Проте, в таких роботах можна побачити і певний парадокс. З одного боку всіма вказується що застосування об'єктного підходу є крок назустріч природним законам психіки людини, а з іншого відомо про програмістів які освоюють цю методологію «насилу», є цілі навчальні керівництва, що вчать «об'єктному мисленню»[5]. Часто доводиться відповідати на питання: чи потрібно привносити об'єктний підхід в навчання, в шкільну програму. Це вимагає додаткового дослідження й такі роботи ведуться на експериментальному рівні (наприклад, Т.Б. Калініна і співавт. [7]).

Висновки. Об'єктно-орієнтованим принципам вже близько півстоліття. За цей час вони змінювалися й уточнювалися, що дозволило говорити про

різні методологічні підходи всередині спільноти математиків і програмістів. У «об'єктних» мов програмування два прабатька. Найбільш поширені мови (C ++, Delphi, C #, Java і ін.) Продовжують об'єктну модель Simula. Мови, що спираються на модель Smalltalk: Python, Ruby. Сьогодні ООП застосовують в різних сферах: від обчислювального експерименту, геометрії й до управління бізнесом. Цілісних психологічних теорій на базі ООП не створено. Є прикладення принципів ООП для вивчення когнітивних процесів, організаційної психології. Дещо більший успіх мають застосування в області педагогічних систем. Проте, підхід цей цікавий і заслуговує на подальше вивчення.

Список використаних джерел

1. Аржанов И.Н. Методика обучения объектно-ориентированному проектированию студентов педагогических вузов / Аржанов И.Н. Авт. дисс...канд. пед. наук, спец. 13.00.02 – теория и методика обучения информатике. – СПб, 2000 – 18 с.
2. Баранова Е.В. Объектно-ориентированное проектирование при обучении современным информационным технологиям / Е.В. Баранова. – СПб : Образование, 2000. – 104с.
3. Баранова Е.В. Теория и практика объектно-ориентированного проектирования содержания обучения средствами информационных технологий. Авт. дисс...докт. пед. наук, спец. 13.00.02 – теория и методика обучения информатике / Е.В. Баранова.– СПб, 2000 – 36 с.
4. Буч Г. ООП / Г.Буч. – М.: Конкорд, 1992. – 519с.
5. Вайсфельд М. Объектно-ориентированное мышление / М. Вайсфельд – СПб : Питер, 2014 .– 304 с.
6. Дорохов В. Н. Математическая психология менеджмента: объектно-ориентированный подход / В. Н. Дорохов // Междунар. конгр. "Маркетинг и пробл. информатиз. предпринимательства", Тезисы. 18–21 нояб., 1996.– СПб, 1996, – с.237–241.



7. Калинина Т.Б. Объектно-ориентированное проектирование в базовом курсе информатики [Электронный ресурс] / Т.Б. Калинина, Н.И. Миндоров, С.В. Русаков. [Пермь]. – Режим доступа : <http://inf.1september.ru/article.php?ID=200601016>

8. Копнов В.А. Объектно-ориентированный подход в менеджменте качества: моногр. / В.А.Копнов, Е.В. Валеева. – Екатеринбург: Урал. гос. лесотехн. ун-т, 2009. – 115 с.

9. Попов Денис Интерфейс vs Классы (2008) [Электронный ресурс] / Денис Попов. – Режим доступа : <http://habrahabr.ru/post/30444/>

10. Степанов А.Г. Объектно-ориентированный подход к отбору содержания обучения информатике / А.Г.Степанов. – СПб : Политехника, 2005. – 287 с.

11. Тей А. Логический подход к искусственному интеллекту / А.Тей, П. Грибомон, Ж.Луи. – М. : Мир, 1990. – 432 с.

12. Тыгу Э.Х. Концептуальное программирование / Э.Х. Тыгу. – М.: Наука, 1984. – 256 с. – с.124–196.

13. Coadd P. Object-oriented patterns. Communication of the ACM. September 1992. – V. 35, No.9. – p. 152–159.

14. Key A. Personal Dynamic Media // IEEE Computer. – 1977. – v. 3. – № 10. – p. 31.

References transliterated

1. Arzhanov I.N. Metodika obucheniya ob'ektno-orientirovannomu proektirovaniyu studentov pedagogicheskikh vuzov / Arzhanov I.N. Avt. diss... kand. ped. nauk, spets. 13.00.02 – teoriya i metodika obucheniya informatike. – Spb, 2000 – 18 s.

2. Baranova E.V. Ob'ektno-orientirovannoe proektirovanie pri obuchenii sovremennym informatsionnyim tehnologiyam / E.V. Baranova. – SPb : Obrazovanie, 2000. – 104s.

3. Baranova E.V. Teoriya i praktika ob'ektno-orientirovannogo proektirovaniya soderzhaniya obucheniya sredstvami informatsionnyih tehnologiy.



Avt. diss...dokt. ped. nauk, spets. 13.00.02 – teoriya i metodika obucheniya informatike / E.V. Baranova.– Spb, 2000 – 36 s.

4. Buch G. OOP / G.Buch. – M.: Konkord, 1992. – 519 s.

5. Vaysfeld M. Ob'ektno-orientirovanoe myishlenie / M. Vaysfeld - Spb : Piter, 2014 . – 304 s.

6. Dorohov V. N. Matematicheskaya psihologiya menedzhmenta: ob'ektno-orientirovannyiy podhod / V. N. Dorohov // Mezhdunar. kongr. "Marketing i probl. informatiz. predprinimatelstva", Tezisy. 18–21 noyab., 1996. – Spb, 1996, – s.237–241.

7. Kalinina T.B. Ob'ektno-orientirovanoe proektirovanie v bazovom kurse informatiki [electronic resource]/ T.B. Kalinina, N.I. Mindorov, S.V. Rusakov. [g. Perm]. – Rezhim dostupa : <http://inf.1september.ru/article.php?ID=200601016>

8. Kopnov V.A. Ob'ektno-orientirovannyiy podhod v menedzhmente kachestva: monogr. / V.A.Kopnov, E.V. Valeeva. – Ekaterinburg: Ural. gos. lesotehn. un-t, 2009. – 115 s.

9. Popov Denis Interfeys vs Klassyi (2008) [electronic resource]/ Denis Popov. - Rezhim dostupa : <http://habrahabr.ru/post/30444/>

10. Stepanov A.G. Ob'ektno-orientirovannyiy podhod k otboru sodержaniya obucheniya informatike / A.G.Stepanov. – Spb : Politehnka, 2005. – 287 s.

11. Tey A. Logicheskiiy podhod k iskusstvennomu intellektu / A.Tey, P. Gribomon, Zh.Lui. – M. : Mir, 1990.–432 s.

12. Tyiugu E.H. Kontseptualnoe programmirovaniye / E.H. Tyiugu. – M.: Nauka, 1984.– 256 s. – s.124–196.

13. Coadd P. Object-oriented patterns. Communication of the ACM. September 1992. –V. 35, No.9. – p. 152-159.

14. Key A. Personal Dynamic Media // IEEE Computer. – 1977. – v. 3. – № 10. – p. 31.

Ця робота ліцензується відповідно до Creative Commons Attribution 4.0 International License.

Авторське право (с) 2016 Stepura Igor Volodymyrovych Степура Ігор Володимирович

Отримано: 21.04.2016

Відрецензовано: 14.05.2016

Опубліковано: 22.06.2016