

дисциплін, спрямованої в русло практичної перевірки дієвості визначених дидактичних умов. Зміст експериментальної програми буде розкрито в наступних публікаціях.

### **Список використаних джерел:**

1. Лобанова А. П. Самостоятельная работа студентов: от контроля и управления к самообразованию / А. П. Лобанова, Н. В. Дроздова // Гуманітарний журнал. – Дніпропетровськ: НГАУ, 2007. – Вип. 34. – С. 54-63.
2. Проблемы диагностики умственного развития учащихся / [под общ. ред. З. И. Калмыковой]. – М.: Педагогика, 1975. – 208 с.

**Гапоненко Н.В.**

*викладач;*

**Бабенко І.Б.**

*викладач,*

*Коледж ракетно-космічного машинобудування  
Дніпровського національного університету  
імені Олесь Гончара*

## **МЕТОДИКА ВИКЛАДАННЯ ДИСЦИПЛІНИ «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ» НА ПРИКЛАДІ РОЗКРИТТЯ ТЕМИ «КОНСТРУКТОРИ ТА ДЕКТРУКТОРИ ПРИ УСПАДКУВАННІ»**

Стаття створена відповідно до тематики програми дисципліни «Об'єктно-орієнтоване програмування» та призначена для розгляду методики викладання теми, яка носить проблемний характер та неоднозначно висвітлюється в різних літературних джерелах. Методика викладання теми містить елементи випереджаючого навчання, використовується компетентнісний підхід, при якому студенти залучаються до проведення логічного аналізу та формулювання і обґрунтування висновків в контексті теми, яка розглядається. При викладанні використовуються наочні матеріали у вигляді рисунків та лістингів з фрагментами програмного коду.

Метою викладання є не тільки поглиблення знань у напрямках вивчення методології об'єктно-орієнтованого програмування, але й виховання компетенцій майбутнього фахівця з розробки програмного забезпечення.

Особливістю запропонованої тематики є проблемна риса. В окремих джерелах тема висвітлена в протилежних фразеологічних зворотах.

Завданням викладача є не тільки розкриття теми, а й доведення важливості розуміння тематики в контексті матеріалу, який викладається. Студент окрім вивчення матеріалу закріплює вміння проводити аналіз, робити логічні висновки, висловлювати і аргументувати власні висновки. Викладач повинен формувати в студентах відповідні вміння, особисті здатності майбутнього фахівця вирішувати певний клас професійних завдань [1]. Необхідно враховувати:

- сучасний зміст навчання, який передбачає як освоєння предметних знань, так і розвиток вмінь студентів, адекватних сучасній бізнес-практиці;
- сучасні методи навчання, які засновані на взаємодії студентів і їх залученні викладачем в навчальний процес;
- сучасну інфраструктуру навчання, яка дозволяє ефективно використовувати переваги різних форм навчання [3].

В процесі навчання в студентах формується схильність до обраної спеціальності, мотивується використання творчого підходу, самостійність. Проводиться актуалізація опорних знань, підкреслюються міждисциплінарні зв'язки, використовується випереджальне навчання.

В сучасному світі швидше зазирнути в інтернет і за лічені хвилини знайти відповідь на питання, аніж прочитати декілька сторінок підручника. Але необхідно навчити студентів не лише користуватися різноплановими джерелами, а і виховати вміння проаналізувати прочитане, зробити висновки.

Конструктори та деструктори при успадкуванні

Механізм успадкування класів дозволяє будувати ієрархії, в яких похідні класи отримують елементи базових класів та можуть доповнювати або змінювати їх. За допомогою успадкування досягається мета – повторне використання програмного коду.

Тема успадкування конструкторів і деструкторів є проблемною з погляду на те, що її не можна вивчати, вирвавши речення-інше з контексту, необхідне розуміння самої концепції успадкування. Приклад розкриття теми наведений далі. «Конструкторы и операция присваивания в производном классе не наследуются, а деструкторы наследуются» [2]. «Хотя конструктор базового класса и наследуется...» [2] «Деструкторы не наследуются» [2].

Успадкування – одна з основоположних концепцій об'єктно-орієнтованого програмування. Фактично тип-нащадок уточнює базовий тип. Порядок конструювання об'єкта похідного класу визначає наступна послідовність: спочатку конструюється базовий підоб'єкт, конструюється похідний об'єкт [3]. При цьому базовий підоб'єкт конструюється повністю і успадковується з урахуванням ключів, що обмежують доступ до його елементів в похідному класі.

Але конструктор базового класу не буде конструктором похідного класу. Тому коли говорять, що конструктори і деструктор з базового класу не успадковуються, мають на увазі те, що вони не існують як конструктори і деструктор похідного класу. Але вони існують як конструктори і деструктор підоб'єкту базового класу.

Це і є відповідь на проблемне питання, питання успадкування конструкторів і деструкторів. І тому, якщо спиратися на цитати буквально, то успадкування цих методів висвітлено неоднозначно, а якщо проаналізувати в контексті успадкування, то все стає зрозумілим.

При створенні похідного класу постає питання, як повинні бути реалізовані конструктори та деструктор. В похідному класі конструктор необхідно писати для ініціалізації власних полів і для передачі значень конструктору базового класу, що виконується явно в списку ініціалізації. При необхідності звільнення пам'яті це доцільно зробити в деструкторі.

Фрагмент відповідного програмного коду наведено на рисунку 1. Зліва наведено використання конструкторів за замовчуванням справа – з параметрами, причому конструктор базового класу викликається явно в списку ініціалізації конструктора похідного класу.

Враховуючи послідовність створення об'єктів в ієрархії, спочатку відпрацьовують конструктори базових класів, починаючи з самого верхнього рівня, потім конструктори об'єктів-елементів класу, і в останню чергу – конструктор класу. При знищенні об'єкта деструктори викликаються в порядку, зворотньому викликам конструкторів.

```

//Специфікація базового класу.
class A {
public:
    int x, y;
}
//Специфікація похідного класу. Відкрите успадкування.
class B : public A {
public:
    int z;
}
... //Фрагмент програмного коду головної функції.
B b;
B.x=1;
B.y=2;
B.z=3;

```

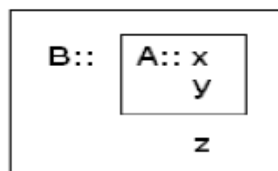
```

//Специфікація базового класу.
class A {
public:
    int x, y;
    A(int p_x, int p_y) : x(p_x), y(p_y) {}
}
//Специфікація похідного класу. Відкрите успадкування.
class B : public A {
public:
    int z;
    B(int p_x, int p_y, int p_z) : A(p_x, p_y), z(p_z) {}
}
... //Фрагмент програмного коду головної функції.
B b(1,2,3);
B.x=B.x+1;
B.y=B.y+2;
B.z=B.z+3;

```

**Рис. 1. Фрагмент програмного коду з відкритим успадкуванням класів**

На рисунку 2 зображено, що об'єкт похідного класу B містить в собі елементи базового класу A.



**Рис. 2. Об'єкт похідного класу B містить в собі елементи базового класу A**

Таким чином, коли говорять про успадкування конструкторів та деструктору базового класу, то розуміють, що вони не існують як відповідні члени об'єкта похідного класу, але існують як відповідні члени підоб'єкта, який відноситься до базового класу. Явний виклик конструктора базового класу передбачається в списку ініціалізації в конструкторі похідного класу, якщо необхідно передати параметри для побудови підоб'єкта базового класу.

Викладач повинен залучити до обговорення проблеми студентів різного рівня підготовки, організувати процес, навчити спілкуванню, надати паузу для осмислення питання, заохотити всіх до логічного аналізу.

Під час викладання теми приділяється увага висвітленню проблеми успадкування конструкторів і деструкторів, в зв'язку з чим студенти навчаються розуміти теоретичні викладки в контексті розгляду теми, а не з цитати, проводити логічний аналіз і аргументувати висновки. Приділяється увага призначенню успадкування, послідовності створення об'єктів класів в ієрархії і яку частину програмного коду слід писати явно. Теоретичні викладки підкреслені відповідними наочними матеріалами. Закріплення матеріалу передбачене під час виконання лабораторних робіт.

Задача викладача – ставити акцент на цілісному сприйнятті, на розумінні й творчості в протизага знанням деталей [4]. Підкреслюється необхідність шукати найбільш ефективні способи реалізації, думати про результати втілення знайденого рішення, аналізувати їх.

Основний принцип освіти – принцип безперервної освіти через все життя на основі саморозвитку [5].

### Список використаних джерел:

- 1 Валлерстайн И. Интеллектуалы в век перехода / И. Валлерстайн // Социология: теория, методы, маркетинг. – 2002. – № 3. – 42-56 с.
- 2 Павловская, Т. А. С/С++. Программирование на языке высокого уровня [Текст] / Т. А. Павловская. – М. – СПб.: ПИТЕР, 2004. – 460 с.
- 3 Довбуш, Г. Ф. Visual C++ на примерах [Текст] / Г. Ф. Довбуш, А. Д. Хомоненко; под ред. А. Д. Хомоненко. – СПб.: БХВ-Петербург, 2008. – 511 с.
- 4 Рудь Марія Компетентнісний підхід в освіті / Марія Рудь // Вісник Львівського університету – 2006 – № 21. – С. 73-82.
- 5 Стив Макконнелл «Личность программиста» [Електронний ресурс]. – Режим доступу до статті: <http://habrahabr.ru/post/24564/> [1a].

**Драгісва Л.В.**

*студент,*

*Ізмаїльський державний гуманітарний університет*

## **ФОРМУВАННЯ КРИТИЧНОГО МИСЛЕННЯ УЧНІВ В УМОВАХ ГРУПОВОЇ РОБОТИ**

Новий освітній стандарт середньої освіти визначив необхідність всесторонньо та гармонійно розвинутої особистості кожного учня [1].

Становлення нової системи освіти вимагає від школи підготовки випускників, здатних швидко адаптуватися до нових життєвих ситуацій, критично мислити, передбачати та вміти розв'язувати створені проблемні ситуації, використовуючи сучасні технології. При цьому важливо бути комунікабельним, вміти колективно працювати в різних галузях, самостійно працювати над розвитком власного інтелекту. Від випускників вищих навчальних закладів реалії сьогодення вимагають нового професійного мислення, компетентності, контактними у різних соціальних групах. орієнтації на внутрішньогрупову діяльність та ін. Означені вимоги пов'язані з необхідністю формування та розвитку критичного мислення. Тож, формування критичного мислення учнів пов'язано не лише з новими задачами, які поставлені перед ЗОШ, а й з тим, що критичне мислення займає суттєве місце в нових, нетрадиційних формах навчання.

Основною метою критичного мислення є розширення мисленевих компетенцій для ефективного розв'язання соціальних, наукових та практичних завдань.

В сучасній школі необхідно перейти до нових ефективних технологій групового навчання, які дозволяють випускникам глибоко адаптуватися до нової системи менеджменту на основі прояву критичного мислення [2, с. 5].

Реалізація групової методики навчання відбувається поступово. На першому етапі діти засвоюють норми спілкування в школі, на уроці, тільки вчаться спілкуватися один із одним.

Поступово діти розуміють, що вони можуть висловлювати думку, при необхідності сперечатися, але тільки нікого не перебиваючи і не ображаючи. Другий етап підготовки – це парна робота. Під час роботи вчитель допомагає парам і фіксує удачі і невдачі в організації. Спочатку вчитель сам каже парам, яким чином вони повинні працювати. Пізніше пари вибирають варіант, виконання завдання якого їм під силу. Пари повинні змінюватися, щоб діти спілкувалися з іншими однокласниками. Але при цьому важливо враховувати психологічні особливості дитини. Не варто садити дітей, які постійно конфліктують один із одним в одну пару. Це може призвести до повної відмови від роботи у парі, а також до конфліктів на