

# СВІТОВІ ОСВІТНІ ТРЕНДИ: НАВЧАННЯ ВПРОДОВЖ ЖИТТЯ В ІНФОРМАЦІЙНОМУ СУСПІЛЬСТВІ





МІНІСТЕРСТВО  
ОСВІТИ І НАУКИ  
УКРАЇНИ



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНА АКАДЕМІЯ ПЕДАГОГІЧНИХ НАУК УКРАЇНИ

УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
імені МИХАЙЛА ДРАГОМАНОВА

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ПЕРЕПІДГОТОВКИ ТА ПІДВИЩЕННЯ КВАЛІФІКАЦІЇ  
УДУ імені МИХАЙЛА ДРАГОМАНОВА

# СВІТОВІ ОСВІТНІ ТРЕНДИ: НАВЧАННЯ ВПРОДОВЖ ЖИТТЯ В ІНФОРМАЦІЙНОМУ СУСПІЛЬСТВІ

МАТЕРІАЛИ

Міжнародної науково-практичної конференції  
присвяченої 190-річчю Університету  
та 50-річчю Інституту

20–21 червня 2024 року

Київ  
Вид-во УДУ імені Михайла Драгоманова  
2024

За загальною редакцією професора **В. П. СЕРГІЄНКА**

**Редакційна колегія:**

- В. П. Андрущенко* – член-кореспондент НАН України, академік Національної академії педагогічних наук України, доктор філософських наук, професор;
- Р. Г. Драпушко* – проректор з науково-педагогічної роботи (адміністративно-господарська діяльність), кандидат філософських наук, доцент;
- В. Г. Лавриненко* – проректор з міжнародних зв'язків, кандидат історичних наук, професор;
- Н. В. Марченко* – кандидат педагогічних наук, доцент;
- Шебень Володимир* – кандидат педагогічних наук, професор (Пряшевський університет, Словаччина);
- В. П. Сергієнко* – доктор педагогічних наук, професор;
- В. М. Слабко* – доктор педагогічних наук, професор;
- Г. М. Торбін* – проректор з наукової роботи, доктор фізико-математичних наук, професор;
- В. І. Федоришин* – доктор педагогічних наук, професор;
- Л. І. Гладка* – кандидат фізико-математичних наук, доцент.

С 24      **Світові освітні тренди: навчання впродовж життя в інформаційному суспільстві:** збірник мат. Міжнародної науково-практичної конф., присвяченої 190-річчю Університету та 50-річчю Інституту. – Київ : Вид-во УДУ імені Михайла Драгоманова, 2024. – 278 с.

До збірника ввійшли матеріали учасників Міжнародної науково-практичної конференції “Світові освітні тренди: навчання впродовж життя в інформаційному суспільстві”, у яких науковці розглядають актуальні питання теорії, методології та практики неперервної освіти в умовах соціальних та інформаційних змін.

Матеріали збірника можуть бути використані науковцями, практиками, здобувачами вищої освіти в галузі психології, педагогіки та дотичних до них наук.

---

---

інформацію, яка стосується міжнародних рейтингів університетів світу, рейтингів навчально-наукових інституцій світу, рейтингів найбільш цитованих університетів, моніторингу суб'єктів науково-видавничої діяльності України. Окремо винесено посилання на рейтинг Українського державного університету імені Михайла Драгоманова у Ukrainian National h-index ranking.

Отже, в результаті модифікації функціоналу та структури веб-сайту Наукової бібліотеки УДУ імені Михайла Драгоманова, яка була здійснена за результатами опитувань та відстеження актуальних інформаційних блоків сайту сторінки “Науковцю” було досягнуто значного покращення у механізмі доступу до наукових ресурсів. Ця ініціатива спрямована на полегшення навігації, забезпечення доступності інформації та стимулювання активного обміну знаннями серед науково-дослідної спільноти.

УДК 378.147:004.42

*Назаренко В. С.,  
кандидат педагогічних наук,  
доцент кафедри інформаційних систем і технологій  
Українського державного університету  
імені Михайла Драгоманова,  
м. Київ, Україна*

### **НАВЧАННЯ СТУДЕНТІВ МЕХАНІЗМІВ РОЗГАЛУЖЕНЬ ТА ЦИКЛІВ НА ОСНОВІ ВИКОРИСТАННЯ КОНЦЕПЦІЇ УМОВНИХ ТА БЕЗУМОВНИХ ПЕРЕХОДІВ**

Програмування, і загалом навички роботи з комп'ютером є дуже важливими у світі. Тому студенти, що вивчають програмування, опановують не тільки сучасну технологію, що стрімко розвивається, але й розвивають логічне мислення, точність формулювань, акуратність, роботу з даними і багато іншого. Але, якщо в процесі навчання не було пропедевтичного вивчення алгоритмізації та програмування, у невмотивованих до різних курсів інформатичних дисциплін студентів виникає незрозуміння необхідності вивчення відповідного розділу навчальної дисципліни. І для викладача стає актуальним завданням – формування мотивації [3] та пізнавального інтересу у процесі навчання програмуванню.

При навчанні програмування студентів молодших курсів (у тому числі спеціальностей, де програмування в будь-якому вигляді не є профільним) можуть виникнути проблеми з розумінням базових конструкцій мови. Для студентів, які раніше займалися програмуванням, особливу складність з базових конструкцій мов імперативного програмування є розгалуження і цикли [1], бо вони порушують природний порядок виконання лінійного коду. У різних мовах програмування, починаючи з нижчого рівня Сі і закінчуючи сучасними мовами високого рівня, є досить багато різних видів циклів та їх форм, у зв'язку з чим може виникати плутанина з послідовністю виконання дій у програмі. Однак на мові асемблера [2] все

---

виглядає дуже схоже, і розуміючи, як це влаштовано зсередини (навіть без знання асемблера), легко не помилитися і масштабувати свої знання про влаштування циклів різними мовами.

У найпопулярніших мовах програмування можна назвати три основні конструкції, що відповідають за нелінійності. Це умовний оператор (if – else у C/C++, if – then – else у Pascal), оператор вибору (switch – case – default і case – of – else відповідно) та різноманітних цикли. Зрозуміло, не у всіх популярних мовах високого рівня є подібні конструкції, наприклад, у мові Python немає оператора вибору та циклу з постумовою (коли спочатку виконується тіло циклу, потім перевіряється умова, таким чином тіло циклу виконається хоча б один раз, це do – while в C/C++, і repeat – until в Pascal), проте їх семантику можна реалізувати за допомогою інших засобів мови.

Зауважимо, що різні альтернативні варіанти запису циклів є, як правило, синтаксичним підсолоджувачем для звичайного циклу з передумовою (while). Код програми можна легко модифікувати, замінюючи один з інших варіантів циклу (ще один приклад – цикл for без аргументів в Go, що є нескінченним циклом без умови) на цикл з передумовою. Зрозуміло, не можна забувати про інструкції break і continue, break з міткою для переходу, зв'язки while – else, for – else (і зрозуміло, їх аналогами інших мов з іншими назвами). Опишемо особливості цих конструкцій.

Інструкція break без аргументів дозволяє вийти з циклу, тобто перейти на наступний рядок коду після завершення блоку циклу. Break з міткою для переходу (Golang, Perl цією можливістю має інструкція last) дозволяє перервати виконання циклу і перейти не на наступний рядок за циклом, а на вказану мітку. Це нагадує оператор goto в Basic, Fortran та інших мовах, але специфічно для циклів. Continue дозволяє перейти на наступну ітерацію циклу, ігноруючи код, який розташовується в тілі циклу після цієї інструкції. Else після циклів у мові Python, наприклад, має наступну семантику: блок else виконується тільки якщо вихід із циклу відбувся після завершення останньої ітерації (тобто не через break).

Окремо варто розглянути for у стилі мови Сі: for (init; expr; increment), де один раз перед початком циклу виконується блок init, далі перевіряється на кожній ітерації істинність умови, і наприкінці кожної ітерації виконується блок increment. Кожен із цих блоків може бути порожнім. Насправді, це аналогічно блоку while з відповідною послідовністю дій.

При нерозумінні виникає питання, що в якій послідовності виконується, коли і куди відбувається вихід із циклу. Також виникають питання, пов'язані з тим, чому після блоку if не виконується блок else в умовному операторі. І у зворотний бік, чому при switch – case потрібно писати break після альтернатив. У промисловому коді трапляються випадки, коли break забувають написати і відбувається “провалювання” в наступну альтернативу. Для контролю цього в стандарті C++17 з'явився атрибут [[fallthrough]], що дозволяє краще контролювати подібну поведінку.

Оскільки не треба пояснювати просте через більш складне та глибоке, створимо форму запису нелінійних переходів, яка використовує такі компоненти:

Мітка (label). Ім'я, яке знаходиться в деякому місці програми і позначає по суті місце у вихідному коді програми, на яке можна перейти. Після імені мітки ставиться двокрапка.

Команда безперечного переходу jmp label. Здійснює перехід на мітку label, тобто, після переходу виконання програми продовжиться з точки label.

---

Команда умовного переходу `ifFalse jmp label`. Здійснюється перехід на мітку `label`, якщо значення попереднього виразу не істинне.

Всі інші конструкції коду, які позначатимемо просто якимись іменами. Префікс `body` означатиме тіло циклу або гілки, `expr` – вираз, результат якого перевірятиметься у `ifFalse`, `increment` – блок коду.

Запишемо у введений вище нотації простий оператор умовного переходу `if – else`.

**Лістинг 1.** Умовне розгалуження в нотації

```
if_branch:
expr      // перевіряємо умову
ifFalse jmp else_branch  // якщо не виконано, переходимо в else
body_if   // якщо виконано, то виконуємо гілку if
jmp end   // і виходимо з конструкції
else_branch:
body_else// інакше виконуємо код гілки else
end:
```

Отже, бачимо, що й умова `expr` не істинна, відбувається умовний перехід на мітку, де починається код `else`-гілки, інакше відбувається виконання коду у гілці `if`. Після її виконання відбувається безумовний перехід на мітку `end`, що знаходиться після завершення розгалуження. Гілка `else` виконується з мітки `else_branch`, після закінчення якої продовжується послідовне виконання, і воно так само продовжить з мітки `end`. У студентів відпадають питання, чому після `if`-гілки не виконується `else`, тому що вони вірять у силу асемблера.

Конструкція оператора вибору є у більш явному вигляді реалізацію ідеї з мітками та умовними та безумовними переходами. У ній є мітки, на які відбувається перехід, причому `break` (або його аналоги), означають перехід на мітку, що розташовується після всієї конструкції. Якщо умова на рівність певному значенню не виконується, відбувається перехід на мітку наступної альтернативи (умовний перехід). Надалі пропонується реалізувати студентам у цій нотації оператор вибору самостійно за аналогією.

Звернемося тепер до запису циклу `while` з `break` та `else`-частиною.

**Лістинг 2.** Цикл `while – else` мовою Python

```
while expr:
body while before_break
if expr_break:
break
body_while_after_break
else:
body_else
```

Цей код буде ідентичним:

**Лістинг 3.** Цикл `while – else` в нотації

```
start:
expr      // умова перебування в циклі
ifFalse jmp else_branch // якщо не виконано, виходимо в else-гілку
body while before break // виконуємо код у тілі циклу
expr break // обчислюємо вираз для виходу
ifFalse jmp end   // якщо воно є істинним, то виходимо
body_while_after_break // якщо ні, то продовжуємо виконання
jmp start // переходимо на наступну ітерацію
```

---

```
else_branch:
body_else      // якщо вийшли не по break, потрапили сюди
end:
```

З реалізації циклу `while` з `break` без `else` замість `else_branch` буде `end`, а реалізації циклу `while` з `continue` перехід відбуватиметься не так на мітку `end`, але в мітку `start`, щоб перейти наступну ітерацію циклу.

Залишилося розібрати `for` в стилі `Ci`, решту пропонується описати студентам для закріплення.

#### **Лістинг 4.** Цикл `for` мовою `Ci`

```
for(init;expr;increment) {
body;
}
```

Цей код буде ідентичним:

#### **Лістинг 5.** Цикл `for` у стилі `Ci` у нотації

```
init
start:
expr          // умова перебування у циклі
ifFalse jmp end // виходимо з циклу
body          // виконуємо тіло циклу
increment     // виконуємо інкремент (як правило, i++ або i--)
jmp start    // переходимо на наступну ітерацію циклу
end:
```

Насамкінець зауважимо, що запис різних циклів і розгалужень в описаній нотації дозволяє наочно показати послідовність виконання команд і наблизити студентів до практичного розуміння, як усе налаштовано насправді.

### ***Література:***

1. Сейдаметова З. С., Манжос Л. О. Мови програмування в навчанні майбутніх програмістів. *Науковий часопис НПУ імені М. П. Драгоманова. Серія 2. Комп'ютерно-орієнтовані системи навчання.* 2010. Вип. 8 (15). С. 35-41.
2. Співаковський О. В. Алгоритмізація та програмування. Енциклопедичне видання : навчально-методичний посібник. К. : ТОВ Редакція "Комп'ютер", 2007. 128с.
3. Шевчук П. Г., Шевчук П. Г. Значення стилю програмування в процесі навчання учнів та студентів. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво* : науковий журнал ; відп. ред. В. Д. Рудь. Луцьк, 2011. № 5. С. 148-150.