



Для другого типу завдань можна запропонувати такі: змоделювати процес побудови кола заданого радіусу; змоделювати рух кульки, кинутої під певним кутом; змоделювати рух більярдної кульки; змоделювати зіткнення кульок тощо.

Для третього типу завдань можна запропонувати такі:

- створити застосунок, у якому є можливість переведення температури зі шкали Цельсія у шкалу Фаренгейта і навпаки;
- створити найпростіший текстовий редактор, у якому можна відкривати, редагувати і зберігати текстові файли;
- створити гру «Камінці баше» чи «Нім» тощо.

Модуль tkinter призначений для створення програм, що забезпечують стандартний інтерфейс користувача. Проте його графічні можливості набагато ширші. Це надає можливість створювати на його основі застосунки різної складності. Треба зауважити, що для розробки ігор зі складною графікою найчастіше використовують додаткові бібліотеки, зокрема, бібліотека Pygame. Однак для простих програм достатньо і того функціоналу, який надає tkinter разом з іншими вбудованими модулями.

#### **Список використаних джерел:**

1. Інформатика для 10-11 класів (профільне навчання). URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/01/10-11-profilniy-riven.docx> (дата звернення 16.06.2024)
2. Інформатика. Навчальна програма вибірково-обов'язкового предмету для учнів 10-11 класів загальноосвітніх навчальних закладів (рівень стандарту). URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/informatika-standart-10-11.docx> (дата звернення 16.06.2024)

## **ПІДГОТОВКА СТУДЕНТІВ ДО УЧАСТІ У ЗМАГАННЯХ З ОЛІМПІАДНОГО ПРОГРАМУВАННЯ**

*Кривонос Олександр Миколайович,  
доцент кафедри комп'ютерних наук та інформаційних технологій,  
кандидат педагогічних наук, доцент  
Житомирський державний університет імені Івана Франка, м. Житомир, Україна  
[krypton@zu.edu.ua](mailto:krypton@zu.edu.ua)*

У світі практично всі галузі певною мірою використовують інформаційні технології. Цифрова економіка потребує зміни підходів до навчання майбутніх фахівців. Важливо переглянути акценти у навчанні фахівців ІТ-сфери, щоб вони не тільки вміли вирішувати стандартні професійні завдання, але й мали знання різних методів критичного мислення та вміння застосовувати їх при розробці алгоритмів та вирішенні нестандартних завдань. Олімпіади з програмування – це перехід від стандартного навчального плану до розширення освітньої програми, збільшення кількості та підвищення якості освоєння основ та базових навичок професії.

Здатність критично мислити та творчо вирішувати нестандартні професійні завдання найбільш яскраво проявляється при вирішенні олімпіадних завдань у галузі алгоритмічного програмування та участі у змаганнях зі спортивного програмування різного рівня. Основна ідея нашої методики спрямована на розвиток професійних навичок майбутніх програмістів під час підготовки команд до участі у змаганнях з алгоритмічного програмування та у процесі самих змагань. Вона включає чотири чинники формування фахового вдосконалення.

Чинники формування фахового вдосконалення, що реалізуються під час підготовки та участі у змаганнях з олімпіадного програмування, включають: формалізацію та алгоритмізацію завдань, програмування мовою високого рівня, тестування та командну роботу.

Проведений аналіз практичної діяльності дозволив розробити систему роботи тренера, виділивши чотири основні етапи підготовки команди до участі у змаганнях з алгоритмічного програмування: підготовчий, базовий, просунутий та професійний. На кожному із цих етапів визначено функції відповідних чинників фахового вдосконалення.

На підготовчому етапі формування фахового вдосконалення відбувається навчання задачам-шаблонам, розглядаються основні прийоми формалізації описових завдань, математично обґрунтовується існування різних підходів їх вирішення, а для реалізації алгоритмів використовуються шаблони кодів. На цьому етапі студентам не потрібно самостійно розробляти тестові завдання; вони перевіряють правильність вирішення завдань готових тестових випадках. Особливість даного етапу полягає у формуванні позитивних емоцій студентів до процесу вирішення нестандартних завдань, що досягається завдяки вибору відповідної педагогічної технології. Формат роботи у цих темах щонайменше важливий, ніж сама тема. Команда створюється стихійно і, як правило, відсутні чіткі ролі.

На базовому етапі ускладнюються типи розв'язуваних завдань, відбувається перехід від задач-шаблонів до задач-алгоритмів. Характеристика конкретної завдання цього залежить від кількості ходів, або підзадач, на які можна розкласти основну задачу. Причому синтез на кожному наступному кроці залежить від результатів аналізу попереднього і формалізований вже лише на рівні методів, які слід застосовувати на кожному наступному кроці. Завдання-алгоритми на базовому рівні мають одноходовий характер і базуються на вивчених задачах-шаблонах. Студентам пропонується самостійно вигадати алгоритм рішення та запрограмувати його, при цьому допускається використання стандартних бібліотек. Починається процес навчання складання тестових завдань для перевірки правильності роботи розробленого алгоритму. Студентам ставиться завдання отримання правильних результатів під час тестування розробленої програми. Основна відмінність цього етапу від попереднього полягає у підвищенні самостійності як на етапі алгоритмізації, так і на етапі розробки тестових завдань. У команді, інтуїтивно самими учасниками або за допомогою тренера, розподіляються ролі та конкретизуються їхні функціональні обов'язки.

Просунутий етап формування фахового вдосконалення пов'язаний з розв'язанням багатоходових задач з алгоритмічного програмування. На цьому етапі студенти самостійно розробляють алгоритми, покращують навички програмування та розглядають питання оптимізації коду. У сфері тестування вони вчаться виділяти

граничні умови під час складання тестових випадків. Особливістю даного етапу є максимальна самостійність у розробці алгоритмів та тестів, а також удосконалення програмного коду. Вектор командної роботи спрямовано розвиток гнучких навичок кожного учасника. У цьому видозмінюються рольові кластери команди: у складі її членів визначається капітан, і всі члени команди намагаються виконувати функції інших ролей.

Останній (професійний) етап фахового вдосконалення пов'язаний із вирішенням нестандартних завдань та розробкою складних алгоритмів. На цьому етапі студенти часто виконують рефакторинг коду, розглядають питання оптимізації часу виконання завдань та мінімізації витрат пам'яті. І тому вони розробляють тестові випадки, враховують дані обмеження. Крім того, на цьому етапі підвищується рівень складності завдань, що вимагає від студентів глибших знань і умінь.

Успішна реалізація запропонованої методики можлива лише за використання різних педагогічних технологій, які змінюються залежно від етапу фахового вдосконалення. Для базового та підготовчого рівнів достатньо лекцій та ігрових методик, які сприяють підвищенню психоемоційного чинника. На базовому рівні, коли розуміються одноходові завдання-алгоритми, доцільно застосовувати технологію проблемного навчання, а також використовувати прийоми та методи критичного мислення.

На просунутому рівні необхідно організувати командну роботу та проводити змагання всередині ВНЗ, у ході яких можуть уточнюватися ролі членів команди. Важливо підтримувати інтерес студентів до цієї діяльності, що сприяє їхньому подальшому розвитку.

На професійному рівні слід забезпечити можливість командам брати участь у всеросійських та міжнародних змаганнях з олімпіадного програмування, що дозволить студентам застосувати свої навички на практиці та отримати цінний досвід.

#### **Список використаних джерел:**

1. Вапнічний С.Д., Жуковський С.С. Проведення шкільних олімпіад з інформатики в умовах пандемії. Вісник Житомирського державного університету імені Івана Франка. Педагогічні науки. 2022. Вип. 3 (110). С. 67-84. [https://doi.org/10.35433/pedagogy.3\(110\).2022.67-84](https://doi.org/10.35433/pedagogy.3(110).2022.67-84)
2. Волошина Т., Глазунова О., Гуржій А., Пархоменко О., Корольчук В. (2020). Платформи та системи автоматизованої перевірки завдань з програмування: аналіз, критерії добору та приклад використання. Електронне наукове фахове видання “ Відкрите освітнє е-середовище сучасного університету”, (8), 154-164. <https://doi.org/10.28925/2414-0325.2020.8.14>
4. Літня школа з програмування : Матеріали лекцій, умови та розбір задач 2017-2019 рр. / За ред. Сергія Вапнічного, Олександра Міци, Сергія Оришича. – Ужгород : Рік-У, 2020. – 336 с.