

НАВЧАННЯ ПОЧАТКІВ ПРОГРАМУВАННЯ НА УРОКАХ ІНФОРМАТИКИ

У статті проаналізовано навчальні матеріали українських науковців, вказано на недостатність матеріалів для формування азів програмування в початківців. Запропоновано поповнити загальний доробок задачами логічного плану та з елементами моделювання, які повинні викликати інтерес у школярів, і разом з тим сприяти розвитку їх логічного мислення. Матеріал містить опис побудови математичних моделей запропонованих задач, слайди з поясненнями та код їх реалізації мовою програмування C++.

Мета. *Розробити окремі компоненти методики навчання початків програмування для учнів середніх шкіл, детально описати задачі, які б викликали у них інтерес до програмування.*

Методологія. *Вивчення та аналіз наукових публікацій, навчально-методичних видань, підбір різнопланових задач з інформатики для зацікавлення учнів з урахуванням досвіду роботи в середній школі, системний підхід до навчання інформатики.*

Наукова новизна. *Підібрано цікаві задачі, зроблено їх детальний опис, описано методику їх навчання з використанням слайдів із поясненнями та коду їх реалізації мовою програмування C++.*

Висновки. *Запропонована методика успішно використовувалась при проведенні початкових занять з учнями Ужгородської загальноосвітньої спеціалізованої школи-інтернату з поглибленим вивченням окремих предметів (УЗСШзПВОП), на літніх та зимових учнівських школах з програмування в Кременчуці та на літніх школах з програмування в Хусті. Багато учнів УЗСШзПВОП почали захоплюватись програмуванням, брали участь в олімпіадах і ставали призерами різних етапів Всеукраїнської учнівської олімпіади з інформатики. Причому за останні три роки щороку хтось з учнів не тільки був у числі переможців олімпіади всеукраїнського рівня, а й виборював призові місця на Європейській юніорській олімпіаді з інформатики. Також вказаний підхід пройшов успішне апробування в умовах дистанційного навчання.*

Ключові слова: *програмування, лінійні конструкції, логічне мислення, мова програмування C++.*

Постановка проблеми

Актуальність роботи. ІТ-галузь є сьогодні мейнстримом розвитку економіки у світі. З кожним роком кількість працівників у ній зростає високими темпами. Зокрема в Україні за останні 5 років їх кількість зросла більше, ніж у два рази, і на сьогодні в Україні налічується понад 212 тисяч працівників ІТ-галузі [1]. З одного боку, це досить велика кількість працівників і досить високі темпи зростання. Але, з іншого боку, у порівнянні із сусідніми європейськими державами кількість програмістів за співвідношенням із кількістю населення в Україні суттєво менша. Наприклад, у Польщі, яка менша за кількістю населення, у цій галузі працює 554 тисячі осіб [2]. Тобто кількість програмістів у співвідношенні з кількістю населення в Україні в два з половиною рази менша, ніж у Польщі. Відповідно, потрібно більше активізувати роботу, пов'язану із заохоченням учнів вибирати професію програміста. Вибір професії учні роблять ще в школі. Тому конче необхідно максимально зацікавити їх на перших уроках з програмування. Відповідно необхідно мати в арсеналі набір якісного матеріалу, який має пробудити інтерес учнів до програмування. Підібрані задачі повинні спонукати учнів до активного думання й пошуку способів знайти розв'язок за невелику кількість кроків.

Аналіз останніх досліджень та публікацій. У зв'язку з великим зацікавленням темами, пов'язаними з навчанням програмування учнів, є значна кількість наукових праць, методичних матеріалів та посібників українських авторів, роботи яких адаптовані для навчання учнів середніх шкіл нашої держави. Зокрема, для формування наборів задач для уроків інформатики, на яких вивчається програмування, є досить великий їх вибір у численних наукових доробках таких українських авторів: Присяжнюк А. В., Вітюк О. В., Лисогор І. А. [3], Милв А. В. [4], Пасіхов Ю. Я. [5], Караванова Т. П. [6–9], Порубльов І. М., Ставровський А. Б. [10], Мельник В. І. [11], Вапнічний С. Д., Зубик В. В., Ребрина В. А. [12], Матвійчук С. В., Жуковський С. С. [13] та інших. Задачі, які наведені в [3] та [13], можна вільно розв'язувати на відомому інтернет-порталі eolump [14], на якому, до речі, розміщено більше 10000 задач

різної складності. Ще одним українським інтернет-ресурсом на якому можна вільно розв'язувати задачі та знаходити навчальні матеріали, є `algotester` [15], який розроблений доцентом Львівського національного університету імені Івана Франка, переможцем фінального етапу студентської олімпіади з програмування під егідою ICPC у 2008 році Білецьким В. М. З методикою розв'язання олімпіадних задач можна ознайомитись у публікаціях провідних українських науковців Горошка Ю. В., Міци О. В., Мельника В. І. [16]. Задачі підвищеної складності за матеріалами міжнародної літньої студентської школи з програмування, яка проводиться щорічно в Ужгороді, разом з теоретичним матеріалом та розв'язками зібрано в книзі [17].

Але серед розглянутих праць практично відсутня інформація про задачі, з яких потрібно починати. Якщо аналізувати матеріали підручників для середніх шкіл різних редакцій, то можна звернути увагу на недостатню кількість матеріалу для зацікавлення учнів займатися програмуванням. Саме ліквідації цієї прогалини і присвячена ця робота.

Мета. Розробити окремі компоненти методики викладення початків програмування для учнів середніх шкіл, детально описати задачі, які б викликали у них інтерес до програмування. Це повинно допомогти вчителю у викладанні азів програмування у середній школі.

Методологія. Вивчення та аналіз наукових публікацій, навчально-методичних видань, підбір різнопланових задач з інформатики для зацікавлення учнів з урахуванням досвіду роботи в середній школі; системний підхід до навчання інформатики.

Наукова новизна. Підібрано цікаві задачі з фабулою щодо навчання азів програмування, зроблено детальний опис їх розв'язування, наведено методику їх викладення з використанням слайдів з поясненнями та наведено код їх реалізації мовою програмування C++.

Результати роботи

Після ознайомлення учнів із загальною схемою програми, командами вводу-виводу та типами даних, настає черга пояснення матеріалу, пов'язаного з

лінійними конструкціями. Тут необхідно мати в арсеналі набір якісного матеріалу, який має пробудити в учнів інтерес до програмування. Підібрані задачі повинні спонукати учнів до активного думання й пошуку способів знайти розв'язок за невелику кількість кроків.

Доцільно запропонувати учням розв'язати без використання команди розгалуження наведені нижче чотири задачі, які, як показує досвід, учні розв'язують із цікавістю.

Задача 1. Наступне парне

Дано ціле число N . Напишіть програму, яка виводить наступне за N парне число. Зі вхідного потоку уводиться єдине ціле число N (від -10000 до 10000). У вихідний потік необхідно вивести єдине ціле число – наступне за N парне.

Приклади

Вхідові дані	Виходові дані
11	12
12	14

Розв'язання

Ця задача, з одного боку, досить проста, з іншого – учням потрібен певний час, щоб знайти одну з відповідей, а їх може бути багато. Серед відповідей, які частіше за все озвучують учні, трапляються такі:

$$- \text{int } M = (N / 2) * 2 + 2;$$

$$- \text{int } M = ((N + 2) / 2) * 2;$$

$$- \text{int } M = N + 2 - N \% 2;$$

$$- \text{int } M = N + 1 + ((N + 1) \% 2) * 2.$$

Звертаємо увагу, що значення N може бути і від'ємне. Тому всі ці відповіді (та й багато інших варіантів) потребують корегування. На рисунку 1 наведено відкорегований розв'язок з врахуванням можливих і від'ємних значень змінної N .

Задача Наступне парне

```
#include<cstdlib>
int main() {
    int n;
    cin>>n;
    n = n + 2 - abs(n)%2;
    cout<<n<<endl;
}
```

Опис змінних

Обчислення результату

Вивід результату

Рисунок 1. Код рішення задачі на C++

Наступна задача повинна навчити учнів поєднувати дві події в одну, щоб розв'язок її сприймався досить легко. Розглянемо її умову.

Задача 2. Кінець уроків

У певній школі уроки починаються о 9.00. Урок триває 45 хвилин, після 1-го, 3-го, 5-го і т.д. уроків перерва 5 хвилин, а після 2-го, 4-го, 6-го і т.д. уроків – перерва 15 хвилин. Вам відомий номер уроку (число від 1 до 10). Потрібно визначити, коли закінчується вказаний урок. Зі вхідного потоку уводиться єдине ціле число від 1 до 10. У вихідний потік необхідно вивести два цілих числа: час закінчення уроку в годинах та хвилинах.

Приклад

Вхідові дані	Виходові дані
2	10 35

Розв'язання

Розв'язок цієї задачі подається у вигляді чотирьох слайдів з детальними коментарями вчителя.

Задача Кінець уроків

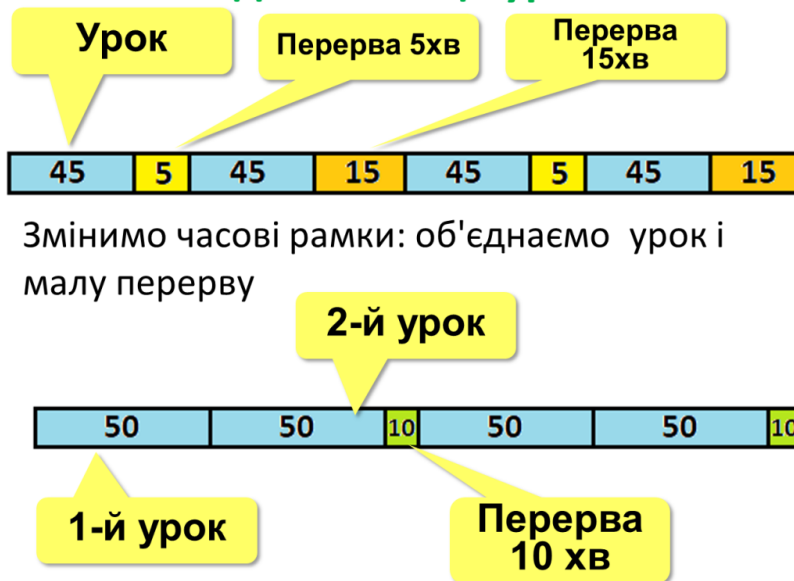


Рисунок 2. Спрощення моделі

Розв'язок задачі може бути легше сприйнятий, якщо виконати дві такі дії:

1) Спростити модель (рис. 2). Спрощення буде полягати в зменшенні кількості задіяних елементів, якими оперуємо. На початку маємо урок тривалістю 45 хвилин і дві різні перерви – 5 та 15 хвилин. Зменшимо кількість даних, якими оперуємо, з трьох до двох. Для цього в моделі приберемо перерви після непарних уроків. Це легко зробити, уявно збільшивши тривалість уроку на 5 хвилин.

Математична модель:

Нехай маємо парну кількість уроків:

$$50n + n/2 * 10 - 15 = 55n - 15$$

50	50	10	50	50	10
----	----	----	----	----	----

Наприклад: $n=4$

Тривалість уроків та перерв:

$$55n - 15 = 55 * 4 - 15 = 205 \text{ хв.}$$

Рис. 3. Математична модель для парної кількості уроків

2) Розглянемо окремо два випадки – при парній і при непарній кількості уроків.

При парній кількості уроків будемо мати (рис. 3):

– тривалість уявно збільшених уроків рівною $50 * n$;

– тривалість перерв у спрощеній моделі $(n/2) * 10 - 15$.

Підсумуємо обидва значення:

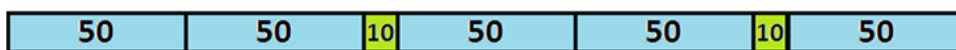
$$50 * n + (n/2) * 10 - 15 = 55 * n - 15.$$

Тобто тривалість занять при парній кількості уроків рівна $55 * n - 15$. В кінці кожного парного уроку є перерва тривалістю 15 хвилин, відповідно останню враховувати не потрібно, тому у формулах тривалості перерв та в кінцевій формулі потрібно відняти 15.

Математична модель:

Нехай маємо **непарну** кількість уроків:

$$50(n-1) + (n-1)/2 * 10 + 45 =$$
$$50n - 50 + 5n - 5 + 45 = 55n - 10$$



Наприклад: $n=5$

Тривалість уроків та перерв:

$$55n - 10 = 55 * 5 - 10 = 260 \text{ хв.}$$

Рис. 4. Математична модель для непарної кількості уроків

При непарній кількості уроків будемо мати (рис. 4):

– тривалість уявно збільшених уроків рівною $50 * (n - 1) + 45$;

– тривалість перерв в спрощеній моделі буде $(n - 1)/2 * 10$.

Підсумуємо обидва значення:

$$50 * (n - 1) + 45 + (n - 1)/2 * 10 = 55 * n - 10.$$

Математична модель:

Об'єднаємо дві умови:

Загальна кількість тривалості n уроків та перерв у хвилинах:

$$m = 55n - 10 - ((n+1) \% 2) * 5;$$

```
int n; cin >> n;
```

```
int m = 55 * n - 10 - ((n + 1) % 2) * 5;
```

```
int h = 9 + m / 60;
```

```
m = m % 60;
```

```
cout << h << " " << m;
```

Рис. 5. Об'єднання обох моделей в одну разом з програмною реалізацією

Звертаємо увагу, що при парній кількості уроків відповіддю буде $55 * n - 15$, а при непарній – $55 * n - 10$. Постає нова задача, яка дуже сильно перегукується з попередньою, – об'єднати ці два результати. Це легко зробити, не використовуючи команду розгалуження. Беремо за основу відповідь $55 * n - 10$ і віднімаємо ще 5, якщо n є парним. У результаті матимемо такий вираз:

$$55 * n - 10 - ((n + 1) \% 2) * 5.$$

Таким чином ми отримали відповідь до цієї задачі у хвилинах. Перевівши в години і хвилини, – отримаємо шукану відповідь згідно з умовою задачі (рис. 5).

Наступна задача повинна ознайомити учнів з азами розв'язання ігрових задач.

Задача 3. Гра

Максимко та Дмитрик грають у цікаву гру. На столі лежать дві купки з M та N горішків. Максим починає завжди першим і хлопці роблять ходи по черзі. За один хід можна зробити одну з таких дій:

- З'їсти один горішок з будь-якої купки.
- З'їсти по одному горішку з обох купок.

– Перекласти один горішок з однієї купки на іншу.

Виграє гру той гравець, що з'їдає останній горішок. Максимко та Дмитрик нескінченно розумні, а тому грають оптимально. Напишіть програму, яка визначає результат гри: виводить 1 – якщо перемаже Максим, 2 – якщо перемаже Дмитро та 0 – якщо гра ніколи не закінчиться.

Зі вхідного потоку уводиться два цілих числа M і N ($1 \leq M, N \leq 2000000000$), що розділені пробілами.

У вихідний потік необхідно вивести число 1, якщо гра завершиться перемогою Максима; число 2, якщо гра завершиться перемогою Дмитрика; число 0, якщо гра ніколи не завершиться.

Приклад

Входові дані	Виходові дані
2 2	2

Розв'язання

При розв'язанні цієї задачі будемо розрізняти два стани позиції в грі залежно від значень M та N – виграшний (В) та програшний (П). Відразу зауважимо, що випадку, коли гра ніколи не завершиться, – не буде. Це твердження поки що неочевидне, але повернемося до нього, розібравши розв'язок цієї задачі.

При розв'язанні будемо користуватись такими двома правилами:

1) якщо всі ходи ведуть у виграшну позицію, то поточна позиція є програшною (П);

2) якщо є хоча б один хід у програшну позицію, то поточна позиція є виграшною (В).

Першою позицією, стан якої легко визначити, є позиція (0, 0). Хто в неї потрапив, той, звісно, програв. Відповідно, позначаємо її як програшну (П). Всі позиції, які можуть у неї потрапити, позначаємо як виграшні (В). Це будуть позиції (0, 1), (1, 0) та (1, 1). Далі звертаємо увагу на позицію (0, 2). З неї можна потрапити у позицію (0, 1), з'ївши один горішок, або у позицію (1, 1), переклавши один горішок. Обидві ці позиції є виграшними. Тому позначаємо

позицію (0, 2) як програшну (П). Позиції, які ведуть у неї, (0, 3), (1, 3) та (1, 2) будемо відзначати як виграшні (В). Симетрично робимо такі ж позначки для позиції (2, 0) як програшної та позицій (3, 0), (3, 1) та (2, 1) як виграшних. Заповнивши за цими правилами квадрат 7 x 7, отримаємо результат, який наведено в таблиці 1.

Таблиця 1

Значення станів позицій при різних значеннях M та N

$M \setminus N$	0	1	2	3	4	5	6	7
0	П	В	П	В	П	В	П	В
1	В	В	В	В	В	В	В	В
2	П	В	П	В	П	В	П	В
3	В	В	В	В	В	В	В	В
4	П	В	П	В	П	В	П	В
5	В	В	В	В	В	В	В	В
6	П	В	П	В	П	В	П	В
7	В	В	В	В	В	В	В	В

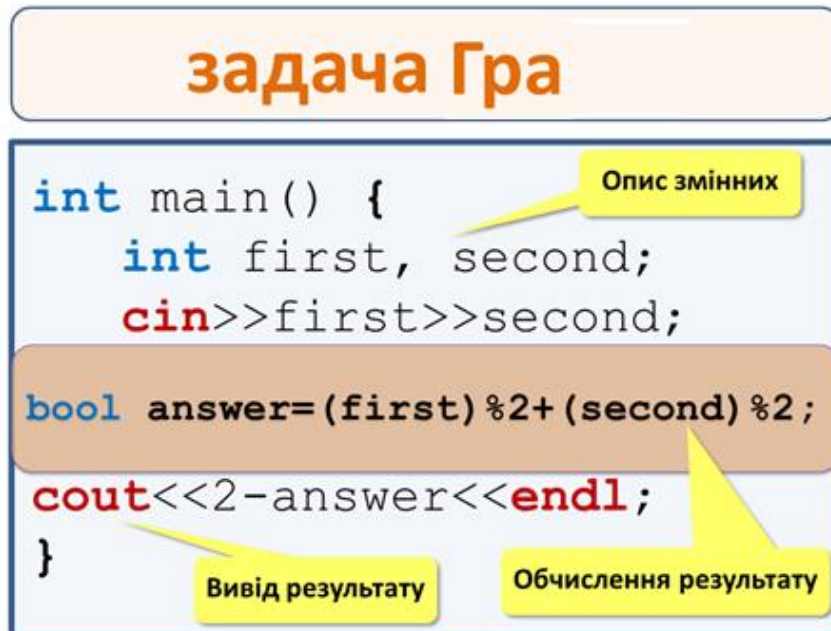
Із таблиці 1 також бачимо, що всі комірки її легко заповнюються згідно з уведеними правилами. Відповідно, гра завжди дійде до завершення. Тому випадку, коли гра ніколи не завершиться, – не буде. Тому розв’язок повинен передбачати лише вивід значень 1 та 2.

Також із таблиці 1 видно повну картину розміщень програшних (П) та виграшних (В) позицій. Легко бачити, що позиція програшна лише тоді, коли обидва значення M та N є парними. Введемо додаткову змінну, яка буде рівна сумі залишків від ділення на 2 значень M та N :

$$answer = M\%2 + N\%2.$$

Значення $answer = 0$ означає, що позиція (M, N) є програшною і, відповідно, хто з неї ходить, той програє. Тобто для неї потрібно вивести 2 (виграє другий гравець). Коли значення $answer = 1$, то це означає, що позиція (M, N) є виграшною і, відповідно, хто з неї ходить, той виграв. Тобто для неї потрібно вивести 1 (виграє перший гравець). Маючи вже пораховане значення

answer, легко скореговуємо відповідь згідно з умовою, вивівши значення $2 - answer$ (рис. 6).



The image shows a code editor window with the following C++ code:

```
int main() {
    int first, second;
    cin >> first >> second;

    bool answer = (first) % 2 + (second) % 2;

    cout << 2 - answer << endl;
}
```

Annotations in yellow callouts point to specific parts of the code:

- "Опис змінних" (Description of variables) points to the variable declarations: `int first, second;`
- "Обчислення результату" (Calculation of the result) points to the calculation of `answer`: `bool answer = (first) % 2 + (second) % 2;`
- "Вивід результату" (Output of the result) points to the output statement: `cout << 2 - answer << endl;`

Рис. 6. Розв'язок задачі "Гра"

Наступною є ще одна задача на моделювання.

Задача 4. Равлик

Равлик повзе по вертикальному стовпі висотою h метрів, піднімаючись за день на a метрів та опускаючись за ніч на b метрів. На який день равлик доповзе до вершини стовпа?

Зі вхідного потоку вводиться три цілих числа h ($1 \leq h \leq 100000$), a ($1 < a \leq 100$) та b ($1 \leq b < a$).

У вихідний потік необхідно вивести єдине ціле число – відповідь на задачу.

Приклад

Вхідові дані	Виходові дані
10	8
3	
2	

Розв'язання

Розв'язок цієї задачі легше уявити, зробивши схематичне зображення всього процесу (рис. 7).

Задача Равлик

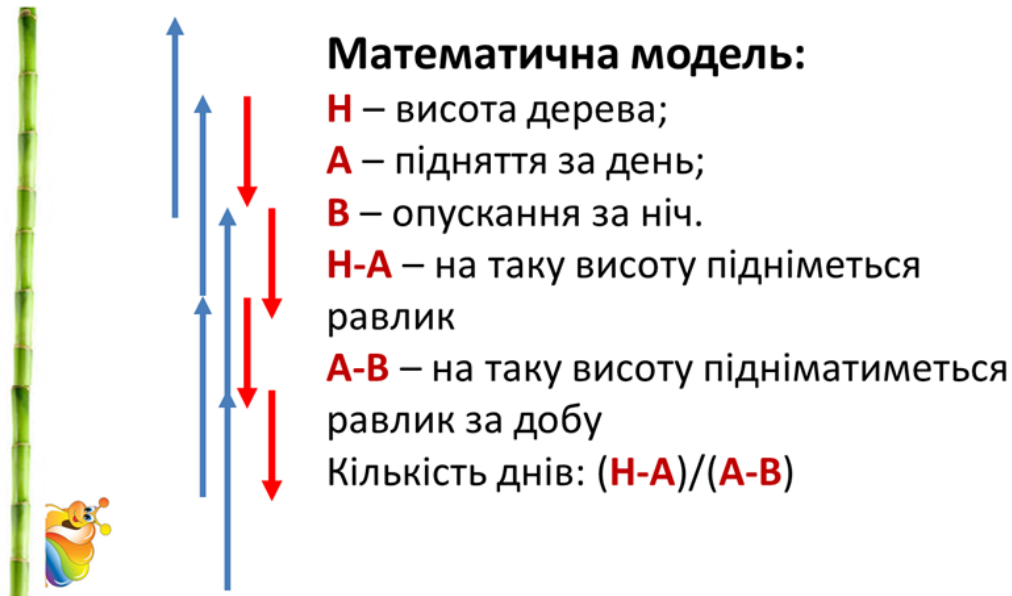


Рис. 7. Модель задачі

Як бачимо, за добу равлик піднімається на $(a - b)$ метрів. Піднімаючись у якийсь із днів на a метрів, він досягне потрібної висоти h . Відповідно, равлик повинен пройти не менше, ніж $(h - a)$ метрів, щоб наступного дня завершити процес підйому. Тому легко визначити кількість днів, коли він досягне висоти принаймні $(h - a)$ метрів. Це буде значення, яке рівне

$$\left\lceil \frac{h-a}{a-b} \right\rceil.$$

У мові програмування C++ функцією, яка повертає найменше ціле, але не менше ніж аргумент, є `ceil()`. Відповідно, вираз `ceil((h - a) / (a - b)) + 1` дає правильну відповідь на поставлену задачу. Плюс один у цьому виразі вказує, що потрібно ще один день, щоб досягнути мети. Повна реалізація мовою програмування C++ наведена на рисунку 8.

Задача Равлик



```
int main()  
{  
    double h,a,b, ans;  
    cin>>h>>a>>b;  
    ans=ceil((h-a)/(a-b))+1;  
    cout<<answer<<endl;  
}
```

Рис. 8. Програмна реалізація

Висновки

Запропонована методика успішно використовувалась при проведенні початкових занять з учнями Ужгородської загальноосвітньої спеціалізованої школи-інтернату з поглибленим вивченням окремих предметів (УЗСШІзПВОП), на літніх та зимових учнівських школах з програмування в Кременчуці та на літніх школах з програмування в Хусті [18]. Багато учнів УЗСШІзПВОП почали захоплюватись програмуванням, брали участь в олімпіадах і ставали призерами різних етапів Всеукраїнської учнівської олімпіади з інформатики. Причому за останні три роки щороку хтось із учнів не тільки був у числі переможців всеукраїнських олімпіад, а й виборював призові місця на Європейській юніорській олімпіаді з інформатики (у 2019 році Олег Навер виборов срібну медаль, у 2020 році Ілля Кривов'яз також виборов срібну медаль [19], а у 2021 році Мартін Лешко виборов бронзову медаль [20]). Також указаний підхід пройшов успішне апробування в умовах дистанційного навчання [21].

Учням, яким легко буде даватися засвоєння матеріалу, пов'язаного з програмуванням, рекомендується брати участь в олімпіадах, кількість яких в Україні зростає. Це і Всеукраїнська учнівська олімпіада з інформатики, і Всеукраїнські юніорська та дівоча олімпіади з програмування, і Всеукраїнська

командна олімпіада з інформатики. Детальна інформація про них наведена на відповідному інтернет-ресурсі [22].

References

1. Скільки IT-спеціалістів в Україні: +29 тисяч за рік згідно з Мін'юстом [Електронний ресурс]. Доступно: <https://dou.ua/lenta/articles/how-many-devs-in-ukraine-2020/>. Дата звернення: Квіт. 12, 2021.

Skilky IT-spetsialistiv v Ukraini: +29 tysiach za rik zghidno z Miniustom [How many IT specialists in Ukraine: +29 thousand per year according to the Ministry of Justice] URL: <https://dou.ua/lenta/articles/how-many-devs-in-ukraine-2020/>.

2. Pół miliona specjalistów IT w Polsce. To problem [Електронний ресурс]. Доступно: <https://crn.pl/aktualnosci/pol-miliona-specjalistow-ict-w-polsce-to-problem/>. Дата звернення: Серп. 19, 2021.

Pol miliona specjalistov IT v Polsce. To problem [Half a million IT specialists in Poland. To problem]. URL: <https://crn.pl/aktualnosci/pol-miliona-specjalistow-ict-w-polsce-to-problem/>.

3. Присяжнюк А.В., Витюк О.В., Лисогор І.А. Олімпіади Житомирської області з інформатики (до 2000 року). Ж.: Соляріс, 1999. 174 с.

Prysiashniuk, A.V., Vytiuk, O.V., Lysohor, I.A. (1999) Olimpiady Zhytomyrskoi oblasti z informatyky (do 2000 roku) [Olympiads of Zhytomyr region in computer science (until 2000)]. Zh.: Soliaris.

4. Милв А.В. Основы программирования в задачах и примерах. Харьков: Фило, 2002. С.397.

Mylyv, A.V. (2002). Osnovi prohrammyrovanyia v zadachakh y prymerakh [Fundamentals of programming in problems and examples]. Kharkov: Fylo.

5. Пасіхов Ю.Я. Всеукраїнські Інтернет олімпіади з інформатики NetOI. В: Універсум. 2006. 150 с.

Pasikhov, Yu.Ia. (2006). Vseukrainski Internet olimpiady z informatyky NetOI [All-Ukrainian Internet Olympiads in Informatics NetOI]. V: Universum.

6. Караванова Т.П. Основи алгоритмізації та програмування: 777 задач з рекомендаціями та прикладами: Навч. посіб. Доп. та випр. К.: Генеза, 2006. 288 с.

Karavanova, T.P. (2006). Osnovy alhorytmizatsii ta prohramuvannia: 777 zadach z rekomendatsiiamy ta prykladamy: Navch. posib. [Fundamentals of algorithmization and programming: 777 problems with recommendations and examples: Tutorial] K.: Heneza.

7. Караванова Т.П. Інформатика. Збірник вправ та задач з алгоритмізації та програмування (процедурне програмування). Навч. посіб. Доп. та випр. Шепетівка: Аспект, 2004. 160 с.

Karavanova, T.P. (2004). Informatyka. Zbirnyk vprav ta zadach z alhorytmizatsii ta prohramuvannia (protsedurne prohramuvannia). Navch. posib. [Informatics. Collection of exercises and tasks on algorithmization and programming (procedural programming). Tutorial]. Shepetivka: Aspekt.

8. Караванова Т.П. Методи побудови алгоритмів та їх аналіз. Необчислювальні алгоритми. Навч. посіб. К.: Генеза, 2006. 224 с.

Karavanova, T.P. (2006). Metody pobudovy alhorytmiv ta yikh analiz. Neobchysliuvalni alhorytmy. Navch. posib. [Methods of constructing algorithms and their analysis. Non-computational algorithms. Tutorial] K.: Heneza.

9. Караванова Т.П. Методи побудови алгоритмів та їх аналіз. Обчислювальні алгоритми. Навч. Посіб. К.: Генеза, 2008. 334 с.

Karavanova, T.P. (2008). Metody pobudovy alhorytmiv ta yikh analiz. Obchysliuvalni alhorytmy. Navch. Posib. [Methods of constructing algorithms and their analysis. Computational algorithms. Tutorial]. K.: Heneza.

10. Порубльов И.Н., Ставровский А.Б. Алгоритмы и программы. Решение олимпиадных задач. К.: Диалектика, 2007. С. 474.

Porublov, Y.N., Stavrovskiy, A.B. (2007). Alhorytmy y prohrammy. Reshenye olymпыadnykh zadach [Algorithms and programs. Solving Olympic problems]. K:Dyalektyka.

11. Мельник В.І. Інформатика. Олімпіадні задачі з розв'язаннями. Х.: Вид. група «Основа», 2010. 159 с.

Melnyk, V.I. (2010). Informatyka. Olimpiadni zadachi z rozv'iazanniamy [Informatics. Olympiad problems with solutions]. Kh.: Vyd. hruppa «Osнова».

12. Вапнічний С.Д., Зубик В.В., Ребрина В.А. Збірник вправ з програмування для слухачів курсів. Розв'язання задач на мові програмування C++. Хмельницький, 2019. 95 с.

Vapnichnyi, S.D., Zubyk, V.V., Rebryna, V.A. (2019). Zbirnyk vprav z prohramuvannia dlia slukhachiv kursiv. Rozv'iazannia zadach na movi prohramuvannia C++ [Collection of programming exercises for course participants. Solving problems in C ++ programming language]. Khmelnytskyi.

13. Матвійчук С.В., Жуковський С.С. Практикум програмування Python / C++ на e-olymp.com (збірник задач з рекомендаціями до їх розв'язання). Житомир: Вид-во ЖДУ, 2019. 235 с.

Matviichuk, S.V., Zhukovskyi, S.S. (2019). Praktykum prohramuvannia Python / C++ na e-olymp.com (zbirnyk zadach z rekomendatsiiamy do yikh rozv'iazannia) [Python / C ++ programming workshop at e-olymp.com (a collection of tasks with recommendations for their solution)]. Zhytomyr: Vyd-vo ZhDU.

14. Інтернет-портал організаційно-методичного забезпечення дистанційних олімпіад з програмування для обдарованої молоді навчальних закладів України e-olymp [Електронний ресурс]. Доступно: <https://www.e-olymp.com/>. Дата звернення: Лист. 17, 2021.

Internet-portal organizacijno-metodychnogo zabezpechennya dystancijnykh olimpiad z programuvannya dlya obdarovanoji molodi navchalnykh zakladiv Ukrainy e-olimp [Internet portal of organizational and methodological support of remote programming competitions for gifted youth of educational institutions of Ukraine e-olimp]. URL: https://www.e-olymp.com/.

15. Платформа автоматичного тестування та проведення змагань зі спортивного програмування з навчальними матеріалами у вільному доступі

algotester [Електронний ресурс]. Доступно: <https://www.algotester.com/>. Дата звернення: Лист. 17, 2021.

Platforma avtomatychnogo testuvannya ta provedennya zmaganiy zi sportyvnoho programuvannya z navchalnyimi materialamy u vilnomu dostupi algotester [Platform for automatic testing and competitions in sports programming with educational materials in free access]. URL: <https://www.algotester.com/>.

16. Горошко Ю.В., Міца О.В., Мельник В.І. Методичні підходи до розв'язування олімпіадних задач з інформатики. *Інформаційні технології і засоби навчання*, 2019, Том 71, №3. С. 40-52.

Horoshko, Yu. V., Mitsa, O.V., & Melnyk, V.I. (2019). Methodychni pidxody do rozvyazuvannya olimpiadnykh zadach z informatyky. Informacijni tehnologiyi i zasoby navchannya [Applying of the general scheme to solving the olympiad task on computer science. Information Technologies and Learning Tools].

17. Літня школа з програмування: Матеріали лекцій, умови та розбір задач 2017-2019 рр. / За ред. Сергія Вапничного, Олександра Міци, Сергія Оришича. Ужгород : Рік-У, 2020. 336 с.

Vapnichnyi, S.D., Mitsa, O.V., Oryshych, S.S. (2020). Litnia shkola z prohramuvannia: Materialy lektsii, umovy ta rozbir zadach 2017-2019 rr. [Summer School of Programming: Lecture materials, conditions and analysis of tasks 2017-2019]. Uzhhorod : Rik-U.

18. Мица О., Орышич С., Заркуа Т., Вапничный С., Горошко Ю. Особенности проведения летней школы по программированию в условиях пандемии. Transactions. Georgian Technical University. Automated Control Systems. Tbilisi. Vol.1.1. № 1(32). 2021. pp. 48–51.

Mitsa, O., Oryshych, S., Zarkua, T., Vapnychnyi, S., Horoshko, Yu. (2021). Osobennosty provedeniya letnei shkoli po prohrammyrovanyiu v usloviakh pandemyu [Features of conducting a summer school on programming in a pandemic]. Transactions. Georgian Technical University. Automated Control Systems. Tbilisi.

19. Одразу два ужгородці представлятимуть Україну на Європейській юніорській олімпіаді з програмування [Електронний ресурс]. Доступно: <http://osvita.uz.ua/odrazu-dva-uzhgorodtsi-predstavlyatymut-ukrayinu-na-yevropejskij-yuniorskij-olimpiadi-z-programuvannya/>. Дата звернення: Лист. 17, 2021.

Odrazu dva uzghorodtsi predstavliatymut Ukrainu na Yevropeiskii yuniorskii olimpiadi z prohramuvannia [Two Uzhgorod residents will represent Ukraine at the European Junior Programming Olympiad]. URL: <http://osvita.uz.ua/odrazu-dva-uzhgorodtsi-predstavlyatymut-ukrayinu-na-yevropejskij-yuniorskij-olimpiadi-z-programuvannya/>.

20. 4 срібні та 4 бронзові медалі вибороли юні українські програмісти на Європейській юніорській олімпіаді [Електронний ресурс]. Доступно: <http://osvita.uz.ua/4-sribni-ta-4-bronzovi-medali-vyboroly-yuni-ukrayinski-programisty-na-yevropejskij-yuniorskij-olimpiadi/>. Дата звернення: Лист. 17, 2021.

4 sribni ta 4 bronzovi medali vyboroly yuni ukrainski prohramisty na Yevropeiskii yuniorskii olimpiadi [4 silver and 4 bronze medals were won by young Ukrainian programmers at the European Junior Olympiad]. URL: <http://osvita.uz.ua/4-sribni-ta-4-bronzovi-medali-vyboroly-yuni-ukrayinski-programisty-na-yevropejskij-yuniorskij-olimpiadi/>.

21. Вапнічний С.Д., Путканадзе Х., Міца О.В., Горошко Ю.В. Організація та аналіз експерименту щодо дистанційного навчання основ програмування учнів віддалених районів. Актуальні питання сучасної педагогіки: творчість, майстерність, професіоналізм: матеріали Міжнародної науково-практичної конференції, Кременчук, 13 березня 2020 р., с. 283-289.

Vapnichnyi, S.D., Putkanadze, Kh., Mitsa, O.V., Horoshko, Yu.V. (2020). Orhanizatsiia ta analiz eksperymentu shchodo dystantsiinoho navchannia osnov prohramuvannia uchniv viddalenykh raioniv. Aktualni pytannia suchasnoi pedahohiky: tvorchist, maisternist, profesionalizm [Organization and analysis of an experiment on distance learning of the basics of programming students in remote

areas. *Current issues of modern pedagogy: creativity, skill, professionalism*]: *Materials of the International scientific-practical conference. Kremenchuk.*

22. Сайт Всеукраїнських олімпіад з інформатики [Електронний ресурс].
Доступно: <https://oi.in.ua/>. Дата звернення: Лист. 17, 2021.

Sait Vseukrainskykh olimpiad z informatyky [Website of the All-Ukrainian Olympiads in Informatics]. URL: <https://oi.in.ua/>.

Vapnichnyi S.

ORCID 0000-0001-8131-0884

Senior Lecturer Of The Department Of Information

Control Systems And Technologies

Uzhhorod National University

(Uzhhorod, Ukraine) E-mail: svapnichny@gmail.com

TEACHING PROGRAMMING BASICS IN COMPUTER SCIENCE

LESSONS

Ukrainian scientists's educational materials are analysed in the article, and the lack of the materials for programming basics formation for beginners is indicated there. It is suggested to fill the general legacy with logical type tasks and the ones with modelling elements, that have to arouse interest in school students and help the development of their logical thinking at the same time. The materials contain the description of constructing mathematical models of suggested tasks, slides with explanation and the code of their realisation in C++ programming language.

Aim. *To create separate components of methodology of teaching programming basics for middle-school students, to describe in details tasks, which could arouse their interest for studying programming.*

Methodology. *Studying and analyzing scientific, educational and methodical publications selection of various exercises in computer science for students's interest based on experience of working in a middle school, systematic approach to teaching computer science.*

Scientific newness. *Interesting tasks were selected, their description was done in details, and the method of teaching them was described using slides with explanation and the code for their realisation in C++ programming language.*

Conclusion. *Suggested methodology was successfully used by students of Uzhhorod General educational specialized boarding school with advanced study of separate subjects (UGESBS) in summer and winter programming schools in Kremenchuk and in summer programming schools in Khust. Many students of the boarding school (UGESBS) started enjoying programming, took part in olympiads and became prizewinners of different stages of All-Ukrainian olympiad in programming. Moreover, for the last three years, every year someone of the students*

was not only the prizewinner, but also a European junior programming olympiad prizewinner. Also, the specified approach passed the test in terms of distance learning.

Keywords: *programming, linear structures, logical thinking, programming language C++.*

Стаття надійшла до редакції 22.11.2021

Рецензент: доктор педагогічних наук, професор Горошко Ю.В.