

УДК 371.2:378

DOI: <https://doi.org/10.24919/2308-4634.2022.259956>

**Тарас Кобильник**, кандидат педагогічних наук, доцент,  
доцент кафедри інформатики та інформаційних систем  
Дрогобицького державного педагогічного університету імені Івана Франка

**Уляна Когут**, кандидат педагогічних наук, доцент,  
доцент кафедри інформатики та інформаційних систем  
Дрогобицького державного педагогічного університету імені Івана Франка

**Оксана Сікора**, кандидат технічних наук, доцент,  
завідувач кафедри інформатики та інформаційних систем  
Дрогобицького державного педагогічного університету імені Івана Франка

**Володимир Жидик**, старший викладач кафедри інформатики та інформаційних систем  
Дрогобицького державного педагогічного університету імені Івана Франка

### **ДЕЯКІ ПРОБЛЕМНІ АСПЕКТИ НАВЧАННЯ ОСНОВ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ У ШКОЛІ**

*У статті розглядаються деякі проблеми навчання основ алгоритмізації та програмування у шкільному курсі інформатики. Нами визначено й проаналізовано такі проблеми: математична підготовка, неперервність змістової лінії основ алгоритмізації та програмування, вибір мови програмування, наявність навчально-методичних матеріалів, мотивація навчання. Кожна з наведених проблем є важливою для навчання основ алгоритмізації. Аналіз проблемних аспектів здійснюється для учнів 5–11-х класів. Перспективним напрямом майбутніх розвідок вбачаємо розробку навчально-методичних матеріалів з основ алгоритмізації та програмування.*

**Ключові слова:** шкільний курс інформатики; основи алгоритмізації та програмування; мова програмування; математична підготовка; мотивація навчання.

*Літ. 11.*

**Taras Kobylnyk**, Ph.D. (Pedagogy), Associate Professor,  
Associate Professor of the Informatics and Information Systems Department,  
Drohobych Ivan Franko State Pedagogical University

**Uliana Kohut**, Ph.D. (Pedagogy), Associate Professor,  
Associate Professor of the Informatics and Information Systems Department,  
Drohobych Ivan Franko State Pedagogical University

**Oksana Sikora**, Ph.D. (Technical Sciences), Associate Professor,  
Head of the Informatics and Information Systems Department,  
Drohobych Ivan Franko State Pedagogical University

**Volodymyr Zhydyk**, Senior Lecturer of the Informatics and Information Systems Department,  
Drohobych Ivan Franko State Pedagogical University

### **SOME PROBLEMALASPECTS OF TEACHING THE FUNDAMENTALS OF ALGORITHMIZATION AND PROGRAMMING IN SCHOOL**

*The article considers some problems of teaching the basics of algorithmization and programming in the school course of computer science. We have identified and analyzed the following problems: mathematical training, continuity of the content line of the basics of algorithmization and programming, the choice of programming language, the availability of teaching materials, learning motivation. Today, there is a tendency to expand programming, and programming languages are becoming simpler and more convenient. As a result, the number of programmers will increase. At the same time, the level of deep understanding of the basics of algorithmization will decrease. This is inevitable, but it is not a reason to abandon the study of modern high-level programming languages. The problem of choosing a programming language for learning the basics of algorithmization in general secondary education is relevant and complex. Based on the analysis of scientific publications and personal experience, the criteria for choosing a programming language as a means of learning the basics of algorithmization are determined: modernity, freeness, clarity, and conciseness. It is necessary to clarify what topics, algorithms need to be studied in schools and how to form a sequence of topics and choose the appropriate teaching methods. It is better to focus on learning algorithms, developing skills for solving various problems that they will be able to use in practice in the future. Therefore, the main emphasis should not be on learning a particular programming language, but on learning*

*the basics of algorithmization, and use a programming language as an aid. It is advisable to choose a programming language that is easy to read and easy to write. This will help increase the level of confidence of students who do not have very good knowledge of the basics of algorithmization and who in the future do not link their professional activities with programming. Each of these problems is important for learning the basics of algorithmization. Analysis of problematic aspects is carried out for students in grades 5–11. We see the development of educational and methodological materials on the basics of algorithmization and programming as a promising direction for future explorations.*

**Keywords:** school course of computer science; basics of algorithmization and programming; a programming language; mathematical training; learning motivation.

**П**остановка проблеми. Важливе місце у шкільній інформатиці займає змістова лінія “Основи алгоритмізації та програмування”. Це наскрізна лінія, яка бере початок ще з другого класу. Досить часто вивчення основ алгоритмізації та програмування викликає в учнів труднощі. Відповідно це може знизити мотивацію до навчання. Тому виникає проблема у виборі таких форм, методів та засобів навчання, зокрема мови програмування, щоб якомога більше зацікавити школярів до вивчення основ алгоритмізації та програмування. Очевидно, що для кожної вікової групи вибір мови та середовища програмування буде відрізнятися.

Як правило, для розв’язування задач з основ алгоритмізації необхідно побудувати її математичну модель. Тут на перший план виходить математична підготовка учнів, вміння логічно й критично мислити.

**Аналіз останніх досліджень та публікацій.** На основі аналізу навчальних програм з інформатики автори [9] відзначають неоднозначність бачення стратегії вивчення програмування у закладах загальної середньої освіти, зокрема, вивчення мов програмування не завжди передбачено у старшій школі. Окрім того, у статті акцентується на виборі мови програмування.

Автори статті [3] звертають увагу на таку проблему послідовності вивчення програмування, зокрема, вони пропонують MIT App Inventor як перехідну ланку між вивченням Scratch та текстових мов програмування високого рівня (C++, Python, Java, C#).

У статті [8] висвітлюються проблеми, пов’язані з навчанням програмування, труднощі психологічного характеру, що супроводжують навчання, та шляхи їх подолання. Зокрема, автори на основі спілкування з вчителями інформатики відзначають тенденцію до зниження інтересу учнів до інформатики як навчальної дисципліни, а, особливо, до основ алгоритмізації та програмування.

Поряд з вибором мови важливу роль відіграє і середовище програмування, оскільки вивчення мови програмування розпочинається з вивчення середовища програмування. Автор відзначає, що

від вибору середовища програмування залежить ефективність вивчення сучасних мов програмування [2].

На важливість математичної підготовки вказує М. Жалдак. На його думку, “...алгоритмізація і програмування – це та сама математика: аналіз умов задачі, побудова математичної (чи інформаційної) моделі задачі, розкладання задачі на підзадачі (за правилами структурного програмування і розкладання алгоритму на блоки за методом “згори донизу”, аналіз алгоритму “знизу догори”) аж до найпростіших, які розв’язуються елементарно” [4, 8].

**Мета статті:** аналіз деяких проблемних аспектів навчання основ алгоритмізації та програмування у шкільному курсі інформатики.

**Виклад основного матеріалу.** На основі власного досвіду та проведеного аналізу наукових праць визначимо такі проблемні аспекти навчання основ алгоритмізації та програмування у школі:

- 1) математична підготовка;
- 2) неперервність змістової лінії основ алгоритмізації та програмування;
- 3) вибір мови програмування;
- 4) мотивація навчання.

**1. Математична підготовка.** Безперечно, для того, щоб добре розв’язувати задачі, пов’язані з алгоритмізацією і програмуванням, треба їх розв’язувати якомога більше від найпростіших до якомога складніших. Для цього в тому числі необхідна і добра підготовка з математики, вміння розв’язувати різні математичні задачі, зокрема з дискретної математики (математична логіка, комбінаторний аналіз, теорія графів, теорія чисел), вміння будувати відповідні математичні та інформаційні моделі й аналізувати їх [5]. Крім того, важливість математичних знань вказує і той факт, що розв’язування значної частини олімпіадних задач з програмування містить математичну основу (див. наприклад [6]).

За результатами міжнародного дослідження PISA 36 % українських школярів не опанували мінімального базового рівня з математики. Такий стан речей призводить до зниження здатності критично та логічно мислити, що є надзвичайно важливим для належного опанування основ

алгоритмізації та програмування. Причинами цього вбачають невідповідність вчителів та невмотивованість учнів [1]. Взагалі це надзвичайно складна проблема шкільної математичної освіти, яка потребує ґрунтовного та всебічного дослідження. Рівень математичних знань впливає також на успішність учнів з інших предметів (не тільки інформатики).

**2. Неперервність змістової лінії основ алгоритмізації та програмування.** Інформатика як навчальна дисципліна у школі вивчається з другого класу. У цій статті ми зупинимось на навчанні інформатики у 5–11-х класах. Однією зі змістових ліній є основи алгоритмізації та програмування. Окремо розглянемо навчання у 5–9-х та 10–11-х класах.

У 5–9-х класах основи алгоритмізації та програмування вивчаються окремим розділом “Алгоритми та програми”, що є одним з багатьох розділів шкільного курсу інформатики. Як наслідок, виникають розриви у вивченні основ алгоритмізації і програмування. Це призводить до того, що учні забувають знання та навички, здобуті під час вивчення матеріалу попереднього року (тобто у 8-му класі забувається, що вивчали у 7-му). Це є надзвичайно складною проблемою, яка вимагає комплексного підходу до вирішення. Одним зі шляхів є виокремлення змістової лінії основ алгоритмізації та програмування в окремому дисципліну. Це знову породжує різні питання, серед яких: яка кількість годин на вивчення? з якого класу починати вивчення? який зміст? тощо. Хоча сьогодні таких радикальних змін шкільна інформатична освіта в найближчому майбутньому не готова.

У 10–11-х класах інформатика вивчається на рівні стандарту та на профільному рівні. Зауважимо, що на рівні стандарту в інваріантній частині не передбачається вивчення питань, пов’язаних з основами алгоритмізації та програмування. Вивчення цих питань можливе тільки у вибіркового модулі “Креативне програмування”.

На профільному рівні інформатика вивчається за двома змістовими лініями:

- 1) основи алгоритмізації та програмування (об’єктно-орієнтоване);
- 2) інформаційні технології.

Програмою профільного рівня [7] передбачено вивчення інформатики з розрахунку 5 год. на тиждень. На профільному рівні питання, пов’язані з алгоритмізацією і програмуванням, вивчаються у розділах “Мова програмування та структури даних” (10 клас), “Алгоритми” та “Парадигми та технології програмування” (11 клас). Тут вже

більше акцент робиться на вивчення мови програмування та технологій програмування.

Згідно з програмою з інформатики (профільний рівень), “вчитель самостійно визначає кількість навчальних годин на вивчення того чи іншого розділу (теми) програми та порядок вивчення тем” [7]. Враховуючи це, пропонуємо вивчати змістові лінії паралельно (наприклад, 3 год. основи алгоритмізації та програмування +2 год інформаційні технології). Так буде забезпечуватися неперервність змістової лінії основ алгоритмізації і програмування.

**3. Вибір мови програмування.** Навчальними програмами не визначається, яку мову програмування обирати. Вибір залишається за вчителем або закладом освіти. Навчання основ алгоритмізації та програмування здійснюють з використанням різноманітних платформ: від Pascal до Java чи C#. Взагалі вибір мови програмування є складною та неоднозначною проблемою. Ще у 2013 р. редакція журналу “Комп’ютер у школі та сім’ї” поставила питання “Яку мову програмування вивчати у школі?”. Така ідея вилилася у кілька публікацій [10;11], у яких знані педагоги висловлювали свої думки з приводу цього питання. Кожен з них наводив переконливі аргументи на користь тієї чи тієї мови програмування, зокрема Pascal, VB, VBA, Python, C/C++, C#, Java тощо.

Ми пропонуємо вибір мови програмування здійснювати, враховуючи вікові особливості школярів, тобто для 5–6-х класів – одна мова, 7–9-х класів – інша, 10–11-х – можлива ще інша.

Крім того, для обрання певної мови програмування необхідно сформулювати критерії вибору. На нашу думку, такими можна визначити:

- сучасність;
- безкоштовний транслятор;
- простий і зрозумілий синтаксис;
- підтримка різних парадигм програмування, зокрема структурне, функціональне та об’єктно-орієнтоване.

Враховуючи вікові особливості та критерії вибору мови програмування, пропонуємо для навчання учнів розділу “Алгоритми та програми” у 5–6-х класах використовувати блочні візуальні середовища програмування (Scratch (<https://scratch.mit.edu/>), Google Blockly (<https://developers.google.com/blockly/>), MIT App Inventor (<https://appinventor.mit.edu>) або аналогічні). Такий вибір зумовлений тим, що у візуальних середовищах не потрібно писати (набирати на клавіатурі) текст програми. Для створення додатка необхідно перетягнути відповідні візуальні блоки. Як правило, можна реалізувати

основні алгоритмічні структури: лінійні, розгалуження, циклічні. Основне завдання використання блочних візуальних середовищ програмування – це зацікавити учнів до вивчення основ алгоритмізації, зробити математику та програмування доступними і цікавими для них. Такий підхід, на нашу думку, якнайкраще буде сприяти підготовці учнів до використання текстових мов програмування у 7–9-х класах.

У 7–9-х класах для вивчення розділу “Алгоритми і програми” ми рекомендуємо обрати Python. Такий вибір обумовлений такими чинниками. Мова Python є сучасною і популярною. Станом на травень 2022 р. займає перше місце в рейтингу мов програмування TIOBE Index (<https://www.tiobe.com/tiobe-index>). Python має відкритий програмний код (Open Source), поширюється за GPL-ліцензією. Синтаксис є простим і зрозумілим. Програми, записані мовою Python, як правило, є коротшими від тих самих програм, записаних на Pascal чи C/C++. Підтримуються різні парадигми програмування, зокрема структурне, функціональне, об’єктно-орієнтоване. Окрім консольного інтерпретатора, для Python розроблено кілька різних середовищ розробки програм, зокрема IDLE – стандартне середовище (входить у дистрибутив). Єдиним недоліком є відсутність візуального середовища розробки програм. Це означає, що, використовуючи стандартне середовище IDLE (описане у шкільних підручниках), не можна створити Windows-форму у режимі конструктора.

У 10–11-х класах на рівні стандарту вивчення вибіркового модуля “Креативне програмування” ми також рекомендуємо проводити з використанням мови Python.

На думку авторів статті [9], перспективним сьогодні у старшій школі вивчати популярні мови C#, Java, Python та C++. На профільному рівні вивчення основ алгоритмізації і програмування можуть бути різні підходи. Перший з них полягає у тому, продовжується вивчати мова Python. При цьому вже не вивчаються основні алгоритмічні структури, а мова Python використовується для створення реальних проектів.

Безумовно, школа не готує програмістів, тим більше це не входить до мети навчання шкільної інформатики. Вивчення основ алгоритмізації і програмування розглядається як фундамент для вивчення мов програмування, які будуть необхідні для навчання в університеті та майбутньої професійної роботи.

**4. Мотивація навчання.** Можна почути думки, що основи алгоритмізації та програмування

не потрібно вивчати у школі, оскільки не всі учні в майбутньому будуть (чи планують стати) програмістами. Так, звичайно, не всі. Так само, як і математиками, філологами, музикантами, художниками тощо. Завдання змістової лінії основи алгоритмізації і програмування полягає у розвитку логічного та формуванні алгоритмічного стилю мислення. Цьому сприятиме вивчення елементів структурного програмування, а вивчення об’єктно-орієнтованого – абстрактного мислення.

**Висновки та перспективи подальших досліджень.**

На основі проведеного дослідження для зменшення негативного впливу на навчання основ алгоритмізації та програмування описаних вище проблем ми пропонуємо.

Для учнів 5–6-х класів доцільно для навчання основ алгоритмізації використовувати блочні візуальні середовища програмування (наприклад, Scratch, Google Blockly, MIT App Inventor). Зауважимо, що у підручниках з інформатики для цих класів, як правило, описується середовище Scratch. Такий вибір якнайкраще сприятиме опануванню текстових мов, які вивчатимуться у наступних класах. У 7–9-х класах рекомендуємо як засіб навчання основ алгоритмізації і програмування обрати мову Python. У 10–11 класах інформатичного профілю можна обрати інші мови (окрім Python), наприклад C++, C# чи Java.

Крім того, для комплексного розв’язання наведених у статті проблем, на нашу думку, змістову лінію основи алгоритмізації та програмування доцільно виокремити в самостійну дисципліну, зміст якої б містив і математичні основи інформатики.

Перспективи подальших досліджень вбачаємо у розробці навчально-методичних матеріалів для навчання основ алгоритмізації та програмування з використанням мови Python з врахуванням її особливостей, зокрема динамічної типізації та надто великої високорівневості.

### ЛІТЕРАТУРА

1. “Математична катастрофа”: чому українські школярі розучились логічно мислити? URL: <https://www.dw.com/uk/математична-катастрофа-чому-українські-школярі-розучились-логічно-мислити/a-51698320> (дата звернення: 10.05.2022).
2. Базурін В. М. Середовища програмування як засіб навчання учнів основ програмування. Інформаційні технології і засоби навчання. 2017. Т. 59, вип. 3. С. 13–27.
3. Власій О., Яремій І., Винничук М. Проблема

послідовності вивчення програмування. *Молодь і ринок*. 2021, №10 (196). С.52–57.

4. Жалдак М. І. Інформатика – фундаментальна наукова дисципліна. Вона має вивчати закони природи, інформаційні процеси і відповідні технології (продовження). *Комп'ютер у школі та сім'ї*. 2010, № 3. С. 7–11.

5. Жалдак М. І. Система підготовки вчителя до використання інформаційно-комунікаційних технологій в навчальному процесі. *Науковий часопис НПУ імені М. П. Драгоманова. Серія 2 : Комп'ютерно-орієнтовані системи навчання*. 2011, №. 11. С. 3–15.

6. Жмурко О.І., Охріменко Т.О. Олімпіади з програмування. Прості задачі. Умань: Візаві, 2020. 298 с.

7. 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 12.05.2022).

8. Семеніхіна О.В., Руденко Ю.О. Проблеми навчання програмувати учнів старших класів та шляхи їх подолання. Інформаційні технології і засоби навчання. 2018. Том 66, №4. С.54–64.

9. Іurchenko A.O., Semenikhina O.V., Khvorostina Yu.V., Udovychenko O.M., Petrenko S.I. Навчання програмувати в старшій школі крізь призму чинних навчальних програм. *Фізико-математична освіта*. 2019. Випуск 2(20), Ч. 2. С. 48–55.

10. Яку мову програмування вивчати у школі. *Комп'ютер у школі та сім'ї*. 2013. №7. С.14–18.

11. Яку мову програмування вивчати у школі. *Комп'ютер у школі та сім'ї*. 2013. №8. С.9–18.

## REFERENCES

1. “Matematychna katastrofa”: chomu ukraïnski shkoliari rozuchylys lohichno myslyty? (2022). [“Mathematical catastrophe”: why did Ukrainian students forget to think logically?]. Available at: <https://www.dw.com/uk/математична-катастрофа-чому-українські-школярі-розучились-логічно-мислити/a-51698320> (Accessed 10 May 2022). [in Ukrainian].

2. Bazurin, V. M. (2017). Seredovysycha prohrumuvannia yak zasib navchannia uchniv osnov prohrumuvannia [Programming environments as a means of teaching students the basics of programming]. *Information technologies and teaching aids*. Vol. 59, No. 3. pp. 13–27. [in Ukrainian].

3. Vlasii, O., Yaremii, I. & Vynnychuk, M. (2021).

Problema poslidoynosti vyvchennia prohrumuvannia [The problem of the sequence of learning programming]. *Youth and market*. No.10 (196). pp.52–57. [in Ukrainian].

4. Zhaldak, M. I. (2010). Informatyka – fundamentalna naukova dystsyplina. Vona maie vyvchaty zakony pryrody, informatsiini protsesy i vidpovidni tekhnolohii (prodovzhennia) [Informatics is a fundamental scientific discipline. It should study the laws of nature, information processes and relevant technologies (continued)]. *Computer at school and family*. No.3. pp. 7–11. [in Ukrainian].

5. Zhaldak, M. I. (2011). Systema pidhotovky vchytelia do vykorystannia informatsiino-komunikatsiinykh tekhnolohii v navchalnomu protsesi [The system of teacher training for the use of information and communication technologies in the educational process]. *Scientific journal of NPU named after MP Drahomanov. Series 2: Computer-based learning systems*. No. 11. pp. 3–15. [in Ukrainian].

6. Zhmurko, O.I. & Okhrimenko, T.O. (2020). Olimpiady z prohrumuvannia. Prosti zadachi [Programming Olympiads. Simple tasks]. Uman, 298 p. [in Ukrainian].

7. Informatyka dlia 10–11 klasiv (profilne navchannia) [Informatics for 10–11 grades (profile training)]. Available at: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/01/10-11-profilniy-riven.docx> (Accessed 12 May 2022). [in Ukrainian].

8. Semenikhina, O.V. & Rudenko, Yu.O. (2018). Problemy navchannia prohrumuvaty uchniv starshykh klasiv ta shliakhy yikh podolannia [Problems of learning to program high school students and ways to overcome them]. *Information technologies and teaching aids*. Vol. 66, No.4. pp.54–64. [in Ukrainian].

9. Iurchenko, A.O., Semenikhina, O.V., Khvorostina, Yu.V., Udovychenko, O.M. & Petrenko, S.I. (2019). Navchannia prohrumuvaty v starshii shkoli kriz pryzmu chynnykh navchalnykh prohrum [Learning to program in high school through the prism of existing curricula]. *Physical and mathematical education*. Issue 2 (20), Part 2. pp. 48–55. [in Ukrainian].

10. Iaku movu prohrumuvannia vyvchaty u shkoli (2013). [What programming language to study at school]. *Computer at school and family*. No.7. pp.14–18. [in Ukrainian].

11. Iaku movu prohrumuvannia vyvchaty u shkoli (2013). [What programming language to study at school]. *Computer at school and family*. No.8. pp.9–18. [in Ukrainian].

Стаття надійшла до редакції 19.04.2022