Навчання основ алгоритмізації

План лекції

1. Поняття алгоритму
2. Мета навчання алгоритмізації
3. Основні положення, на яких повинен будуватися сучасний підхід до вивчення основ алгоритмізації
4. Методи навчання класичних алгоритмів
5. Етапи навчання основ алгоритмізації

Вивчення інформатики формує **елементи операційного стилю мислення, який полягає в умінні:**

- **формалізувати задачу;**
- **виділити в ній логічно самостійні частини;**
- **визначити зв'язки цих частин;**
- **спроектувати алгоритм розв'язування за допомогою технологій "згори-донизу" та "знизу-догори";**
- **дібрати якомога ефективніший шлях отримання розв'язку;**
- **інтерпретувати та аналізувати результати.**

Саме тому формування такого операційного стилю мислення є одним із завдань навчання взагалі і курсу інформатики зокрема. Навчання основ алгоритмізації і програмування забезпечує умови для реалізації таких завдань.

Поняття алгоритму. Термін «алгоритм» походить від середньоазіатського міста Хорезм. За наших часів поняття алгоритму було узагальнено, і словом "алгоритм" стали позначати

опис будь-якої послідовності дій. Поняття алгоритму є одним із фундаментальних у сучасній математиці й інформатиці.

Алгоритм - це точний і зрозумілий опис послідовності дій над заданими об'єктами, що дозволяє отримати кінцевий результат;

скінчена послідовність вказівок, формальне виконаній яких дозволяє за обмежений час отримати розв'язок задачі;

певна інструкція для виконавця, яка маже бути задана різними способами — словами, формулами, послідовністю обчислювальних операцій чи логічних дій.

Але не кожна інструкція може бути алгоритмом.

Алгоритм повинен відповідати певним вимогам, **мати такі властивості:**

масовість (має бути придатним для багатьох задач, що належать до певного класу);

детермінованість (визначеність) кожна команда не повинна допускати двоякого тлумачення, кожний крок алгоритму повинен бути точно визначеним;

дискретність (процес, який визначається алгоритмом, повинен мати дискретний (перервний) характер, тобто являти собою послідовність окремих завершених кроків);

результативність (кожна дія повинна приводити до цілком певного результату);

формальність (будь-який виконавець, здатний сприймати та виконувати вказівки алгоритму (навіть не розуміючи їх змісту), діючи за алгоритмом, може виконати поставлене завдання);

скінченність (скінчена кількість кроків).

До слова «алгоритм» близькі за значенням слова: спосіб, рецепт.

Алгоритми в інформатиці - це не тільки рецепти розв'язання

задач. Їх розробляються з метою автоматизації дій виконавця. Алгоритм розв'язаній однієї й тієї самої задачі може бути поданий по-різному.

Алгоритми складаються з орієнтацією на певного виконавця алгоритму: дресированої тварини, людини, автомата, ПК.

До алгоритму мають входити команди, які виконавець може виконати, і неприпустимі команди, які він не може виконати. У кожного виконавця є свій кінцевий набір команд, які для нього зрозумілі і можуть бути виконані. Цей набір називають системою команд виконавця. Користуючись системою команд, виконавець може виконувати алгоритм формально, не вникаючи у зміст поставленої задачі дія виконавця вимагається лише виконання послідовності дій, передбачене алгоритмом. Коли алгоритм зрозумілий конкретному виконавцю, кажуть, що такий алгоритм має властивість певності. Завдяки певності багатократне виконання одного алгоритму різними виконавцями за тих самих вихідних умов приведе до однакових результат Для отримання конкретного результату не допускаються довільні дії з боку виконавця.

В алгоритмі неприпустимі вказівки типу «Додати дві-три ложки цукру» або «Зняти з вогню через кілька хвилин». Потрібно утикати також ситуацій, коли після виконання чергової команди виконавцю незрозуміле, яка команда має виконуватися наступною.

Найкращими є ті алгоритми, які забезпечують розв'язання широкого кола задач (наприклад алгоритм Евкліда, алгоритми виконання арифметичних дій). Про такі алгоритми кажуть, що вони мають властивість масовості. Вони дозволяють розв'язувати (і неодноразово) не одну конкретну задачу, а багато однотипних задач.

Структура алгоритму:

Назва алгоритму (арг1, арг2, ..., аргN)

Опис змінних:

арг1 – опис першого аргументу

арг2 – опис другого аргументу

...

аргN – опис N-го аргументу

Опис результату:

рез – опис того, що повертає алгоритм

Початок

Опис допоміжних змінних:

зм1 – опис першої допоміжної змінної

зм2 – опис другої допоміжної змінної

...

змN – опис N-ї допоміжної змінно

Тіло алгоритму:

крок 1 – опис першої дії

крок 2 – опис наступної дії

...

крок N – опис N-ї дії

Кінець

Мета навчання основ алгоритмізації - навчити основних способів організації операцій і даних, а також застосування базових алгоритмічних конструкцій при складанні описів алгоритмів розв'язування різноманітних задач.

Під час вивчення основ алгоритмізації основна увага насамперед повинна приділятися:

- **виявленню загальних закономірностей і принципів алгоритмізації;**
- **основним етапам розв’язування задач сучасних інформаційних технологій;**
- **аналізові поставленої задачі, методам формалізації та моделювання реальних процесів та явищ;**
- **добору виконавця поставленої задачі, виходячи з того, що він є також певним об'єктом із притаманними йому властивостями та набором допустимих операцій, які слід аналізувати з метою правильного та ефективного їх використання;**
- **методам та засобам формалізованих описів дій виконавця, сучасним засобам їх конструювання та реалізації за допомогою комп'ютера.**

Однією з проблем, яка постає перед учителями при вивченні цього розділу, є поєднання досить консервативної алгоритмічної лінії курсу з динамічними та сучасними лініями виконавця, формалізації та моделювання, інформаційних технологій.

Алгоритмізація як розділ інформатики, який вивчає процеси створення алгоритмів, традиційно відноситься до теоретичної інформатики внаслідок свого фундаментального характеру. Завдяки розвитку інформаційних технологій, і зокрема технологій програмування, з'являється можливість у межах розділу "Основи алгоритмізації та програмування" ознайомити учнів з загальнонауковими поняттями інформатики і в той же час

формувати та розвивати вміння та навички, необхідні користувачеві під час роботи з сучасним програмним забезпеченням, тобто з'являється можливість зробити цей розділ містком між теоретичною та практичною інформатикою.

Розділ "Основи алгоритмізації та програмування" шкільного курсу інформатики має неабияке методологічне значення. Він розкриває важливість алгоритмів, їх роль у функціональному зв'язку понять "інформація-алгоритм-комп'ютер", що визначають процес автоматичного опрацювання інформації. На прикладах демонструється можливість формального виконання алгоритму, елементарність дій, що задаються на виконання виконавцеві за кожною вказівкою алгоритму. Тим самим підкреслюється можливість передавання виконання формально описаного алгоритму виконавцеві-машині, тобто можливість автоматизації діяльності людини на основі алгоритмів. А вивчення алгоритмічної мови дозволяє познайомити учнів з формалізованим записом алгоритмів, тим самим розширити їхні уявлення про засоби описування алгоритмів

Сучасний підхід до вивчення основ алгоритмізації повинен будуватися на таких основних положеннях:

1.Процес вивчення основ алгоритмізації слід орієнтувати на використання комп'ютера як дидактичного засобу навчання.

2.Метою вивчення основ алгоритмізації є виділення змісту реального об'єкта - алгоритму - і правил його побудови, а не вивчення конкретної алгоритмічної мови. Алгоритмічна мова – це лише один із багатьох засобів формального подання алгоритмів.

Поняття алгоритму відіграє провідну роль у формуванні операційного мислення та уявлень учнів про можливість автоматизації різних видів діяльності людини. Введення і розвиток у шкільному курсі інформатики поняття алгоритму та вивчення основних властивостей алгоритмів дозволяє показати учням одну з найважливіших характеристик алгоритмів - формальність, чисто механічний характер діяльності людини при їх виконанні, що є основою виконання такого роду операцій за допомогою комп'ютера, тобто основою автоматизації виконання таких операцій.

Таким чином, етап опанування поняття "алгоритм" є першим етапом формування в учнів уявлень про автоматичне опрацювання різноманітних даних за допомогою комп'ютера. Варто також звернути увагу і на внутрішні предметні зв'язки - **алгоритм можна трактувати як повідомлення про те, як слід розв'язувати задачу, подане у вигляді формально описаного впорядкованого набору вказівок про те, які і в якому порядку слід виконувати операції для того, щоб розв'язати задачу.** Слід звернути увагу учнів на те, що якщо поміняти порядок виконання вказівок, то задача швидше за все не буде розв'язана або ж буде розв'язана зовсім інша задача. Володіння поняттям алгоритму є одним із найважливіших компонентів інформаційної культури.

Поняття "алгоритм" належить до числа фундаментальних математичних понять і є об'єктом дослідження спеціального розділу математики - теорії алгоритмів.

У змісті навчання основ алгоритмізації можна виділити такі компоненти:

- **навчання відомих алгоритмів і їх використання;**

- **навчання класичних алгоритмів;**
- **навчання побудови описів алгоритмів як з використанням, так і без використання відомих алгоритмів.**

Але для вирішення вказаних завдань потрібна навчальна алгоритмічна мова, яка б надавала можливість описувати алгоритми за єдиним набором правил. Оволодіння алгоритмічною мовою один з найважливіших моментів навчання основ алгоритмізації у школі.

На думку А.П.Єршова, **навчальна алгоритмічна мова виконує дві основні функції у навчанні алгоритмізації.** По-перше. її застосування дозволяє стандартизувати, надати єдиної форми описам всіх алгоритмів, які розглядаються в шкільному курсі інформатики, що важливо для розуміння суті алгоритмізації, формування уявлень про властивості алгоритмів. По-друге, вивчення навчальної алгоритмічної мови є пропедевтикою вивчення мов програмування. Простота конструкцій навчальної алгоритмічної мови і правил їх використання дозволяє успішно застосовувати цю мову на початковому етапі навчання програмування. Основні з цих конструкцій і правил лежать в основі багатьох мов програмування. Тому освоєння навчальної алгоритмічної мови дозволить надалі легко перейти до використання реальних мов програмування.

Теоретичною базою навчання вже відомих алгоритмів і їх використання є: теорії Ж.Піаже, М. Паська, С.Пейперта, які об'єднує наступна ідея: для того, щоб оволодіти алгоритмом, потрібно описати його: а) будь-якою з "своїх" мов (Ж.Піаже, М. Пась); б) мовою "Черепашки" (С.Псйперт).

Наведемо один із **варіантів класифікації мов, які можуть бути покладені в основу вивчення та застосування для навчання основ алгоритмізації:**

- **природна мова (словесний опис алгоритму);**
- **мова графічних схем;**
- **структура програми;**
- **шкільна навчальна алгоритмічна мова;**
- **мови програмування.**

Класифікацію виконавців алгоритмів можна подати так:

• людина;

• комп'ютер, що працює на основі використання компіляторів(інтерпретаторів) мов різних рівнів.

Звідси випливає, що для вивчення відомих алгоритмів і їх використання необхідно передусім вибрати виконавця і вивчити систему його вказівок (мову для запису алгоритмів), тобто набір операцій, які може виконувати такий виконавець у зв'язку з задачею, яку необхідно розв'язати.

Таким чином, можна сформулювати **метод вивчення відомих алгоритмів: учням повідомляється алгоритм, описаний деякою мовою, а їм необхідно:**

1) записати алгоритм алгоритмічною мовою або мовою конкретного виконавця (якщо алгоритм повідомлено в словесній формі);

2) виконати алгоритм, використовуючи набір операцій конкретного виконавця.

До класичних алгоритмів можна віднести алгоритми таких типів:

1)для створення, зміни і руйнування інформаційних структур(лінійні списки, дерева, графи);

2)напівчисельні алгоритми - алгоритми теорії чисел, арифметика багаторазової точності (за Д.Кнутом);

3)генерації псевдо випадкових чисел (за Д.Кнутом);

4)пошуку і впорядкування (за Д.Кнутом);

5)пошуку підрядка в рядкові (за Д.Кнутом);

6)стиснення інформації;

7)побудови графічних примітивів.

Враховуючи обмеженість навчального часу, всі класичні алгоритми неможливо вивчити на уроках інформатики, з ними слід знайомити учнів на факультативах або при організації профільного навчання.

Практика свідчить про доцільність використання такого методу навчання класичних алгоритмів:

1.Учням пропонується задача для програмування, яка легко розв'язується за допомогою деякого класичного алгоритму (сам алгоритм учневі не повідомляється).

2.Учень розв'язує задачу, записуючи алгоритм, розроблений самостійно.

3.Після цього вчитель повідомляє класичний алгоритм, описаний рідною мовою або мовою програмування (але в ньому спеціально пропущені деякі рядки програми).

4. Учень реалізовує алгоритм і робить висновок про міру ефективності своєї реалізації (за часом виконання, за довжиною опису).

Наведемо **деякі міркування щодо добору вправ для навчання класичних алгоритмів.**

1. Вправи на тестування (процес пошуку помилок у, програмі). Учневі надається текст програми, в якому описано класичний алгоритм, і словесний опис алгоритму. Потрібно побудувати систему тестів для його перевірки.

2. Вправи на оптимізацію. Учневі для роботи пропонується програма (з можливими помилками). Потрібно виявити помилки без її реалізації на комп'ютері і оптимізувати програму за часом виконання.

При навчанні побудови (відкриття) алгоритмів перш за все необхідно разом з учнями з'ясувати питання: в чому полягає процес побудови алгоритму.

Побудувати алгоритм - означає: 1) продумати план деякої майбутньої діяльності; 2) зафіксувати його за допомогою деякої системи позначень, так званою формальною мовою.

Досвід свідчить, що доцільно сумістити навчання основ алгоритмізації і навчання основ програмування, не відмовляючись від опису алгоритмів рідною мовою, тобто опису схеми розв'язування задачі (ще **Е. Дейкстра** вказував, що **"найбільш важлива перевага професійного програміста, крім математичних здібностей - це вільне володіння рідною мовою"**).

Як правило написанню програми деякою мовою програмування передує розробка алгоритму. Алгоритм можна записувати навчальною алгоритмічною мовою або відповідною реальною мовою програмування. Передбачається, що мова, яка

використовується в таких цілях, містить правила опису основних управляючих конструкцій структурного програмування, зокрема виклику процедур, що дозволяє реалізувати структурний підхід до розробки алгоритмів. У цьому випадку мова програмування не є самостійним об'єктом вивчення, а виступає лише як система позначень, з використанням якої ведеться обговорення проблем і шляхів їх розв'язування.

Доцільно виділити такі **етапи навчання основ алгоритмізації:**

1-й етап. Вступ до алгоритмізації. Алгоритми над найпростішими типами даних. Проектування алгоритмів "згори-донизу". Базові алгоритмічні конструкції: послідовне виконання вказівок, цикл, розгалуження. Побудова алгоритмів методом покрокової деталізація з використанням раніше описаних.

2-й етап. Навчання класичних алгоритмів (раніше описаних) та їх використання.

Зазначимо, що **при навчанні основ алгоритмізації потрібно виділяти три класи вправ**

1) задано текст задачі і алгоритм її розв'язування, потрібно виконати алгоритм і з'ясувати особливості роботи за алгоритмом (вправи на тестування);

2) задано лише опис алгоритму, потрібно виконати алгоритм і встановити його призначення (вправи на "відгадування" формулювання задачі);

3) задано лише призначення алгоритму, потрібно написати текст алгоритму - скласти алгоритм і виконати його - протестувати (вправи на складання алгоритму).

Висновки:

Розуміння алгоритмів та вміння їх створювати – це ключ до розвитку логічного мислення та успішного програмування. Алгоритми допомагають структурувати рішення задач, роблячи їх більш ефективними та послідовними.

Основна мета навчання алгоритмізації – це розвиток в учнів аналітичного та алгоритмічного мислення. Вміння працювати з алгоритмами дозволяє вирішувати різноманітні завдання, від простих до складних, у різних галузях знань.

Навчання алгоритмам має бути побудоване на інтеграції новітніх технологій, використанні практичних завдань, що стимулюють творче мислення. Важливо акцентувати увагу на реальних прикладах застосування алгоритмів та їхньої ролі у розвитку штучного інтелекту, автоматизації та інших актуальних сфер.

Використання різноманітних підходів, таких як наочне моделювання, практичні вправи, крокове виконання та оптимізація існуючих алгоритмів, дозволяє учням краще зрозуміти та засвоїти основні концепції.

Процес навчання алгоритмізації проходить кілька ключових етапів: від ознайомлення з поняттям алгоритму до практичного використання та удосконалення алгоритмів. Ефективне навчання вимагає поступового нарощування складності та включення учнів у процес розробки власних алгоритмів.