**Розробка уроку**

**Тема:** розв’язування задач методом їх розбиття на підзадачі  
  
**Мета:**

* ознайомити з методом покрокової деталізації та послідовного уточнення алгоритму;
* сформувати вміння розробляти алгоритми «зверху донизу»;
* навчити аналізувати алгоритм розв’язання задач за її інформаційною моделлю;
* запровадити поняття підпрограми.

**Хід уроку**  
  
1. Повторити означення понять, виділених жирним шрифтом.  
  
**Алгоритм** — *це запис скінченої послідовності вказівок, виконання яких призводить до розв'язання певної задачі.*  
  
**Вказівка** (алгоритму) — *це спонукальне речення, що вказує, яку дію має виконати виконавець алгоритму.*  
  
**Виконавець** (алгоритму) — *це жива істота (людина або тварина) або автоматичний пристрій (робот, електронна обчислювальна машина тощо), спроможна діяти відповідно з алгоритмом.*  
  
**Система вказівок виконавця** — *це множина (сукупність) всіх вказівок, які може виконувати даний виконавець.*  
  
**Середовище виконання** алгоритму — *об'єкти, з якими працює виконавець у процесі виконання алгоритму.*  
  
**Властивості алгоритму**: *дискретність, визначеність, виконуваність, скінченність, результативність, масовість, ефективність.*  
  
**Дискретність** (латинською discretus — розділений, розривний) алгоритму означає, що *виконання алгоритму зводиться до виконання окремих дій (кроків) у певній послідовності. Причому, кожну вказівку алгоритму виконують за скінченний проміжок часу.*  
  
**Визначеність (однозначність)** означає, що *алгоритм однозначно визначає порядок дій виконавця, результат цих дій і не потребує додаткового тлумачення.*.  
  
**Виконуваність** означає, що *алгоритм, призначений для певного виконавця, може містити лише вказівки, які входять до системи вказівок цього виконавця.*  
  
**Скінченність** означає, що *виконання алгоритму закінчиться після скінченної (можливо, досить великої) кількості кроків і за скінченний час для довільних вхідних даних.*  
  
**Результативність** алгоритму означає, що *після закінчення виконання алгоритму обов’язково:*

* *або одержують розв'язок-результат відповідно до поставленої мети;*
* *або встановлюють відсутність таких розв'язків (множина розв'язків порожня);*
* *або встановлюють неможливість розв'язання задачі.*

**Масовість** алгоритму означає, що *алгоритм можна застосувати до цілого класу однотипних задач, для яких спільними є умова та хід розв’язування та які відрізняються лише початковими (вхідними) даними. Наприклад, алгоритмом дій, складеним для одного касира, можуть успішно скористатися всі касири супермаркету. А програмою пошуку коду і підрахунку суми вартостей товарів, придбаних покупцем, — усі комп'ютери супермаркету*  
  
**Ефективність** алгоритму *описує час виконання і об'єм ресурсів, необхідних для виконання алгоритму*: чим менше часу (часова ефективність) і ресурсів (просторова ефективність), тим ефективність вища.  
  
**2. Вивчення нового матеріалу**  
  
**Вступ**  
*Як Ви вважаєте, чи залежить ступінь деталізації алгоритму від того, на якого виконавця орієнтовано виконання даного алгоритму? Поміркуйте, наприклад, як розробляють конструкцію сучасного теплоходу, автомобілю або літаку. Чи можливо розв’язати конструкторську задачу без поступового заглиблення в деталі?*  
  
**Задача 1**. Скласти алгоритм дій на цілий день, користуючись алгоритмами *Гігієнічні процедури, Ранок, Вечір, Школа.* Порівняти з демонстраційним розв'язанням.

**Алгоритм Гігієнічні процедури**

1. Сходити до вбиральні.
2. Почистити зуби.
3. Умитися.
4. Витертися рушником.

**Алгоритм Ранок**

1. Встати о 7-й годині.
2. Виконати алгоритм *Гігієнічні процедури*.
3. Виконати гімнастичні вправи.
4. Поснідати.
5. Одягнутися.
6. Вийти з дому о 8-й годині.

**Алгоритм Школа**

1. Доки не закінчилися уроки іти на наступний урок згідно з розкладом.
2. (Після завершення уроків) іти додому.

**Алгоритм Вечір**

1. Повернутися зі школи додому після уроків.
2. Перевдягнутися.
3. Пообідати.
4. Якщо погода хороша, то попрацювати в саду.  
   Інакше піти в бібліотеку, поміняти книжки й повернутися додому.
5. Зробити уроки.
6. Повечеряти.
7. Якщо є цікава телепередача, то подивитися телевізор.  
   Інакше почитати книжку.
8. Роздягнутися.
9. Виконати алгоритм *Гігієнічні процедури*.
10. Лягти спати.

**І спосіб розв'язання**  
Перепишемо 6 вказівок алгоритму *Ранок*, додамо 2 вказівки алгоритму *Школа* і 9 вказівок алгоритму *Вечір*. Отримаємо алгоритм з 18 вказівок.  
  
**Алгоритм День**

1. Встати о 7-й годині.
2. Виконати алгоритм *Гігієнічні процедури*.
3. Виконати гімнастичні вправи.
4. Поснідати.
5. Одягнутися.
6. Вийти з дому о 8-й годині.
7. Доки не закінчилися уроки іти на наступний урок згідно з розкладом.
8. (Після завершення уроків) іти додому.
9. Повернутися зі школи додому після уроків.
10. Перевдягнутися.
11. Пообідати.
12. Якщо погода хороша, то попрацювати в саду.  
    Інакше піти в бібліотеку, поміняти книжки й повернутися додому.
13. Зробити уроки.
14. Повечеряти.
15. Якщо є цікава телепередача, то подивитися телевізор.  
    Інакше почитати книжку.
16. Роздягнутися.
17. Виконати алгоритм *Гігієнічні процедури*.
18. Лягти спати.

**ІІ спосіб**  
  
**Алгоритм *День***

1. Виконати алгоритм *Ранок.*
2. Виконати алгоритм *Школа*
3. Виконати алгоритм *Вечір*.

Переваги ІІ способу очевидні. Тут головний алгоритм *День* складається лише з трьох вказівок виклику допоміжного алгоритму (*Ранок, Школа і Вечір*).  
  
**Головний алгоритм** — *це такий алгоритм, виконання якого веде до досягнення основної мети.*  
  
**Допоміжний алгоритм**призначений для досягнення проміжної мети.  
  
Розглянемо головний алгоритм *День.* Він викликає три допоміжні алгоритми. Частини, з яких складається алгоритм, називаються *модулями.*

|  |
| --- |
|  |
|  |  |  | Ранок | — | Гігієнічні процедури |
|  |  | / |  |  |  |
| День |  | — | Школа |  |  |
|  |  | \ |  |  |  |
|  |  |  | Вечір | — | Гігієнічні процедури |

Вище подано модульну структуру алгоритму *День*. Вона відображає, які допоміжні алгоритми використовує основний алгоритм, але (у поданому вигляда) не описує (не вказує чітко)послідовності їх виконання.  
  
Великі алгоритми зазвичай проектують так. Спочатку аналізують умову задачі. Складають загальний план її розв'язування. Якщо задача складна, то її розбивають на декілька простіших підзадач. Проектують модульну структуру алгоритму: описують призначення головного та допоміжних алгоритмів, дають їм назви. Потім деталізують (створюють, уточнюють) необхідні допоміжні алгоритми для розв'язування підзадач. Маючи допоміжні алгоритми, записують головний алгоритм, який складатиметься з команд викликів допоміжних. Отже, великі алгоритми утворюють з готових модулів (блоків) подібно до того, як будинки будують з готових блоків, а машини збирають з окремих деталей.  
  
**Задача 2**.  
Скласти алгоритм дій на тиждень. Тиждень складається з п'яти робочих днів, для кожного з яких підходить алгоритм *День*, а також суботи та неділі. Модульну структуру алгоритму *Тиждень* подано на рисунку нижче.

|  |
| --- |
|  |
|  |  |  |  |  | Ранок | — | Гігієнічні процедури |
|  |  |  |  | / |  |  |  |
|  |  | День |  | — | Школа |  |  |
|  | / |  |  | \ |  |  |  |
| Тиждень | — | Субота |  |  | Вечір | — | Гігієнічні процедури |
|  | \ |  |  |  |  |  |  |
|  |  | Неділя |  |  |  |  |  |

**Алгоритм Тиждень**

1. Доки не настала субота, виконати алгоритм *День*.
2. Виконати алгоритм *Субота*.
3. Виконати алгоритм *Неділя*.

Цей алгоритм стане визначеним, якщо будуть складені допоміжні алгоритми *Субота, Неділя.* Для розв'язання задачі потрібно два кроки (рівні, етапи) деталізації. На першому кроці деталізують алгоритми *День, Субота, Неділя,* а на другому деталізують або використовують алгоритми *Ранок, Школа, Вечір.*  
  
Описаний метод називають *проектуванням алгоритму «зверху-вниз» з покроковою деталізацією (уточненням) допоміжних алгоритмів*.  
  
Метод покрокової деталізації конструювання алгоритмів не враховує явно конкретні особливості поставленої задачі та вибір певного виконавця. Але сукупність вказівок вибраного виконавця істотно впливає на ступінь деталізації алгоритму та на його структуру. Допоміжний алгоритм (як і будь-який інший алгоритм) повинен мати лише один вхід і один вихід. Потрібно щоб усі вказівки допоміжного алгоритму входили до системи вказівок обраного виконавця. У реальному житті допоміжні алгоритми можуть виконувати навіть інші виконавці.  
  
Процес роботи комп’ютера полягає у виконанні *програм, тобто наборів вказівок у визначеному порядку.* Писати такі програми – складна справа. Раніше для цього програміст повинен був пам’ятати не лише всі комбінації нулів та одиниць двійкового коду кожної вказівки, але й двійкові коди адрес даних, використаних під час виконання програми. Щоб полегшити роботу програмістів, було розроблено багато мов програмування високого рівня, які в наочнішому (для людини) вигляді подавали послідовність дій комп’ютера. Такі алгоритмічні мови опису побудованих алгоритмів, призначених для виконання комп’ютерами, називаються *мовами програмування.* Легше написати програму такою мовою програмування, яка наближена до людської, а перекладання з цієї мови мовою машинних кодів доручити комп’ютеру. Допоміжні алгоритми, що природнім чином виникають при використанні методу покрокової деталізації, записують за допомогою функцій (і процедур для Pascal). Ці можливості було передбачено вже у перших мовах програмування високого рівня.  
  
**3. Закріплення вивченого матеріалу**

**Усно дайте відповідь на запитання**

1. Опишіть метод «покрокової деталізації».
2. Що таке допоміжний алгоритм?
3. Що таке головний алгоритм?

**4. Домашнє завдання**

Опрацювати конспект уроку, знати відповідь на питання 1-3.