

**Натисніть тут, щоб
купити книгу на сайті
або замовляйте за телефоном:
(0352) 51-97-97, (067) 350-18-70,
(066) 727-17-62**

ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ

1. ІНФОРМАЦІЙНА (МАТЕМАТИЧНА) МОДЕЛЬ

1.1 Основні етапи розв'язування прикладної задачі з використанням ЕОМ

Розв'язування задач у будь-якій діяльності — це завжди одержування певних результатів — результатів обчислень, побудови, роботи тощо.

ЕОМ є універсальним пристроєм для розв'язування різноманітних навчальних і виробничих задач. Їх універсальність полягає у можливості виконання практично всіх алгоритмів розв'язування задач у будь-якій предметній діяльності. Необхідним для цього стає опис постановок завдань і методів їх розв'язування мовою математики, а потім вираження їх у формі програм.

Етапи розв'язування задач на ЕОМ:

1. Математична постановка задачі.
2. Визначення методів розв'язування.
3. Складання сценарію роботи з ЕОМ.
4. Конструювання алгоритму.
5. Переведення алгоритму у програму.
6. Введення і випробування програми.
7. Одержання результатів розв'язування.

При постановці задачі необхідно визначити і перелічити всі вихідні дані і дані, які необхідно знайти. Відповісти на запитання: за яких умов можна одержати потрібні результати, а за яких ні? Визначити, які результати будуть вважатися правильними?

На другому етапі необхідно не тільки вибрати спосіб одержання результатів, а й вибрати оптимальний метод розв'язування задач, який би давав найбільш правильну відповідь. Правильність розв'язування задач на ЕОМ перш за все залежить від правильності вибраного методу розв'язування.

Для написання сценарію програми необхідно продумати правила введення даних в ЕОМ, форму подання інформації користувачам, реакцію машини

на вказівки і дані, які вводяться. Сценарії повинні визначати правила роботи користувачів з ЕОМ.

За складеним сценарієм і описом методу складаємо алгоритм розв'язування задачі.

Програмування (написання програм за наявності алгоритмів) — кодування алгоритмів вибраною мовою програмування.

Тестування — процес підготовки, виконання програми та аналіз результатів з метою виявлення помилок. Введення і випробування створеної програми здійснюємо за заздалегідь підготовленим планом. Програму виконують на ЕОМ з різними значеннями аргументів, які б забезпечували перевірку всіх можливих умов, при яких може виникнути помилка. Оцінюємо правильність одержаних результатів.

Після закінчення тестування програми виводимо правильні результати на екран монітора або на принтер. На основі отриманих результатів будують таблиці, графіки, аналізують і пояснюють у термінах вихідних даних задачі.

1.2 Поняття інформаційної (математичної) моделі. Побудова моделі

Успішно розв'язувати задачі можна тільки при чіткому і водночас однозначно визначенні вимог до кінцевих результатів. Розпливчастість і невизначеність формулювань може привести до різного тлумачення умов і, як наслідок — розбіжності в оцінці правильності результатів.

Точні постановки задач — це можливість забезпечити однакове розуміння цих задач різними людьми. Точність визначень і формулювань має бути такою, щоб вони не допускали двозначного тлумачення, а найголовніше, щоб за ними можна було однозначно міркувати, чи є запропоновані розв'язки правильними. Така точність формулювань і тверджень характерна для математики. А оскільки ЕОМ — це математичні пристрої для опрацювання даних, то постановка задач, що розв'язуються обчислювальними машинами, також повинна виражатися у математичній формі.

Опис найбільш суттєвих властивостей об'єктів і явищ, які досліджуються в задачі за допомогою математичних формул і рівнянь, називається побудовою математичної моделі цього об'єкта. Математична модель дає можливість звести розв'язування реальної задачі до вирішення математичної задачі. Саме цей факт лежить в основі застосування математики у пізнанні законів і їх практичного застосування.

4.2.2 Опис алгоритмів із розгалуженням мовою програмування

1. Скласти програму обчислення значення функції:

$$Y = \begin{cases} x, & \text{якщо } x < 0; \\ 5, & \text{якщо } 0 \leq x < 5; \\ 5x, & \text{якщо } x \geq 5. \end{cases}$$

Розв'язок.

```

Program PFUNCTION;
Var X,Y:real;
begin
  write('X=');
  readln(X);
  if X<0 then Y:=X
    else if (0<=X) and (X<5)
      then Y:=5
      else Y:=5*X;
  writeln('X=',X,' Y=',Y);
end.

```

2. Скласти програму розв'язування квадратного рівняння.

$$ax^2 + bx + c = 0.$$

```

Program KVADRAT;
Var A,B,C,D,X1,X2:real;
begin
  writeln('Введіть коефіцієнти a, b, c :');
  write('a=');
  readln(A);
  write('b=');
  readln(B);
  write('c=');
  readln(C);
  D:=sqrt(B)-4*A*C;
  if D<0 then writeln('Рівняння має комплексні корені')
    else
      begin
        X1:=(-B-sqrt(D))/(2*A);

```

```

X2:=(-B+sqrt(D))/(2*A);
writeln('Корені рівняння :');
writeln('X1=',X1);
writeln('X2=',X2);

end;

end.

```

3. Скласти програму розв'язування системи двох лінійних рівнянь, що має єдиний розв'язок.

Розв'язок лінійної системи рівнянь виду

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}, \text{ якщо } \Delta = a_1b_2 - a_2b_1$$

можна знайти за формулами

$$X = \frac{c_1b_2 - c_2b_1}{\Delta}, \quad Y = \frac{a_1c_2 - a_2c_1}{\Delta}.$$

```

Program SISTEMA;
Var A1,B1,C1,A2,B2,C2,D,X,Y:real;
begin
  writeln('Введіть коефіцієнти системи :');
  write('a1=');
  readln(A1);
  write('b1=');
  readln(B1);
  write('c1=');
  readln(C1);
  write('a2=');
  readln(A2);
  write('b2=');
  readln(B2);
  write('c2=');
  readln(C2);
  D:=A1*B2-A2*B1;
  if D<>0 then
    begin
      X:=(C1*B2-C2*B1)/D;
      Y:=(A1*C2-A2*C1)/D;
      writeln('Розв'язок системи :');

```

```

        writeln('x',x:5:2,' y',y:5:2);
    end
else writeln('Система не має розв'язку');
end.

```

4. Скласти програму знаходження розв'язку лінійного рівняння
 $ax + b = 0$.

Розв'язання.

```

Program LINRIVN;
Var A,B,X:real;
begin
    writeln('Введіть коефіцієнти a, b :');
    write('a=');
    readln(A);
    write('b=');
    readln(B);
    if A <> 0 then X := -B/A
    else if B = 0 then
        writeln('x може бути довільним числом')
    else writeln('Рівняння не має розв'язків');
    writeln('x=' ,X);
end.

```

5. Скласти програму знаходження найбільшого спільного дільника (НСД) двох натуральних чисел M і N.

Розв'язання.

```

Program PNSD;
Var M,N,X,Y,NSD:integer;
begin
    writeln('Введіть дійсні числа:');
    write('m=');
    readln(M);
    write('n=');
    readln(N);
    X:=M;
    Y:=N;
    while X <> Y do
        if X > Y then X := X - Y
        else Y := Y - X;

```

```

NSD:=X;
writeln('НСД(‘,M,’’,N,’)’,NSD);
end.

```

4.2.3 Практична робота №5

ТЕМА: Створення та реалізація програм із розгалуженням.

МЕТА: Навчитись складати та реалізовувати програми з використанням вказівок розгалуження.

ЗАВДАННЯ ДЛЯ ПРАКТИЧНОЇ РОБОТИ.

1. Вивчити можливості мови програмування Паскаль з реалізації обчислювального процесу із розгалуженням.
2. Відповісти на контрольні питання.
3. Скласти програму розв'язання задачі згідно із завданням.
4. Виконати складену програму.
5. Підготувати і використати тести (кількість тестів дорівнює кількості віток розгалуження) для перевірки правильності функціонування програми.
6. Оформити звіт з практичної роботи.

Зразок виконання практичної роботи

ПРАКТИЧНА РОБОТА №5

ТЕМА: Створення та реалізація програм із розгалуженням.

МЕТА: Скласти програму обчислювального процесу із розгалуженням.

ЗАВДАННЯ. Обчислити податок із заробітної плати в залежності від її величини за формулою:

$$P = \begin{cases} \bar{p} = 0, & \text{якщо } z \leq 17; \\ p = 0.1 * z, & \text{якщо } 17 < z < 85; \\ p = 0.15 * (z - 85) + 6.80, & \text{якщо } z \geq 85 \end{cases}$$

Програма розв'язування даної задачі мовою програмування Паскаль.

```

Program PROBOTA5;
Var P,Z:real;
begin
  writeln('Введіть значення величини заробітної плати :');
  write('z=');
  readln(Z);
  if Z <= 17 then P:=0

```

11. ПРОЦЕДУРИ КЕРУВАННЯ ЗВУКОВИМИ СИГНАЛАМИ

Для відтворення звуку в Турбо Паскалі використовуються процедури модуля Crt в такій послідовності.

Sound

Sound(I) — процедура активізує звукові засоби ЕОМ. Цілочисельне значення I вказує частоту звучання звуку в герцах. Звук вказаної частоти буде звучати до тих пір, доки не буде відмінений процедурою NoSound.

Delay

Delay(MS) — виконує затримку виконання програми на вказане число мілісекунд. MS — число мілісекунд, протягом якого буде звучати звуковий сигнал.

NoSound

NoSound — відміна звуку. Відміняється звуковий режим, заданий процедурою Sound.

Наприклад.

```
Program Dzvuk;  
Uses Crt;  
begin  
    Sound(500);  
    Delay(2000);  
    NoSound;  
end.
```

В даному випадку звуковий сигнал частотою 500 Гц буде звучати протягом 2 секунд (2 000 мілісекунд).

Програма DemoBird демонструє можливості використання звукових процедур.

```
Program DemoBird;  
Uses Crt;  
var  
    I:integer;
```



```
procedure Bird;
begin
  I:=2000;
  While I < 3000 do
    begin
      Sound(I+1);
      Inc(I);
    end;
  NoSound;
end;
begin
  repeat
    for I:=1 to 20 do
      begin
        Sound(Random(100) + Random(500) + 3100);
        Delay(8);
        NoSound;
        Delay(30);
      end;
    Bird;
    Randomize;
    Delay(Random(600)) until KeyPressed;
    NoSound;
  end.
```

ЗМІСТ

| | |
|-------------|---|
| ВСТУП | 3 |
|-------------|---|

ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ

1. ІНФОРМАЦІЙНА (МАТЕМАТИЧНА) МОДЕЛЬ

| | |
|--|---|
| 1.1 Основні етапи розв'язування прикладної задачі з використанням ЕОМ | 4 |
| 1.2 Поняття інформаційної (математичної) моделі. Побудова моделі | 5 |
| 1.3 Запитання, завдання, тести | 7 |

2. АЛГОРИТМИ

| | |
|--|----|
| 2.1 Алгоритми. Властивості алгоритмів. Форми подання алгоритмів. Виконавець алгоритму | 8 |
| 2.2 Базові структури алгоритмів | 9 |
| 2.3 Метод покрокової деталізації. Конструювання алгоритмів «зверху донизу» | 11 |
| 2.4 Навчальна алгоритмічна мова. Правила описання алгоритмів навчальною алгоритмічною мовою | 12 |
| 2.5 Величини. Основні характеристики величин | 16 |
| 2.6 Вказівка присвоювання | 17 |
| 2.7 Запитання, завдання, тести | 18 |

МОВА ПРОГРАМУВАННЯ ПАСКАЛЬ

3. МОВА ПРОГРАМУВАННЯ ПАСКАЛЬ. СИСТЕМА ПРОГРАМУВАННЯ

| | |
|--|----|
| 3.1 Поняття програми. Мова програмування. Переклад програм з однієї мови на іншу. Поняття про транслятори, систему програмування | 22 |
| 3.2 Поняття про середовище програмування Turbo Pascal | 23 |
| 3.3 Практична робота №1 | 25 |
| 3.4 Запитання, завдання, тести | 25 |
| 3.5 Алфавіт мови програмування. Величини та їх опис мовою програмування. Змінні та константи | 27 |
| 3.6 Запитання, завдання, тести | 29 |
| 3.7 Ідентифікатори величин. Типи величин | 30 |
| 3.6 Запитання, завдання, тести | 34 |

| | | |
|-------|--|-----|
| 3.9 | Вказівки введення й виведення. Вказівка присвоювання | 36 |
| 3.10 | Запитання, завдання, тести | 42 |
| 3.11 | Стандартні типи змінних. Опис типів змінних | 43 |
| 3.12 | Опис програм за правилами мови програмування | 44 |
| 3.13 | Практична робота №2 | 46 |
| 3.14 | Запитання, завдання, тести | 47 |
| 3.15 | Набір функцій та операцій, що визначені для кожного із стандартних типів. Арифметичні вирази. Пріоритет операцій | 49 |
| 3.16 | Запитання, завдання, тести | 53 |
| 3.17 | Переклад лінійних алгоритмів з навчальної алгоритмічної мови на мову програмування | 57 |
| 3.18 | Практична робота №3 | 58 |
| 4. | ВКАЗІВКИ ПОВТОРЕННЯ ТА РОЗГАЛУЖЕННЯ | |
| 4.1 | Вказівки повторення і розгалуження | 61 |
| 4.1.1 | Вказівка повторення | 61 |
| 4.1.2 | Переклад циклічних алгоритмів із навчальної алгоритмічної мови на мову програмування | 66 |
| 4.1.3 | Практична робота №4 | 67 |
| 4.1.4 | Запитання, завдання, тести | 69 |
| 4.2 | Вказівка розгалуження | 72 |
| 4.2.1 | Опис вказівки розгалуження мовою програмування | 72 |
| 4.2.2 | Опис алгоритмів із розгалуженням мовою програмування | 78 |
| 4.2.3 | Практична робота №5 | 81 |
| 4.2.4 | Запитання, завдання, тести | 84 |
| 5. | ТАБЛИЧНІ ВЕЛИЧИНИ. МАСИВИ | |
| 5.1 | Табличні величини та їх опис мовою програмування | 87 |
| 5.2 | Алгоритм знаходження суми й добутку елементів таблиць | 89 |
| 5.3 | Складання алгоритмів та програм на опрацювання табличних величин | 91 |
| 5.4 | Алгоритми пошуку в таблицях елементів із деякою властивістю | 95 |
| 5.5 | Алгоритми впорядкування табличних величин | 96 |
| 5.6 | Складання алгоритмів та програм на опрацювання табличних величин | 99 |
| 5.7 | Практична робота №6 | 102 |
| 5.8 | Запитання, завдання, тести | 104 |

| | |
|--|-----|
| 6. ЗВЕРНЕННЯ ДО АЛГОРИТМІВ І ФУНКЦІЙ | |
| 6.1 Підпрограми. Типи підпрограм. Опис підпрограм мовою програмування. Звернення до підпрограм | 108 |
| 6.2 Створення програм із використанням підпрограм | 114 |
| 6.3 Практична робота №7 | 115 |
| 6.4 Запитання, завдання, тести | 117 |
| 7. РЯДКОВІ ВЕЛИЧИНИ | |
| 7.1 Вказівки і функції опрацювання рядкових величин | 120 |
| 7.2 Складання та реалізація програм опрацювання рядкових величин | 122 |
| 7.3 Практична робота №8 | 123 |
| 7.4 Запитання, завдання, тести | 126 |
| 8. ГРАФІЧНИЙ РЕЖИМ РОБОТИ ЕОМ | |
| 8.1 Вказівки для роботи у графічному режимі. Створення графічних примітивів | 129 |
| 8.2 Практична робота №9 | 134 |
| 8.3 Запитання, завдання, тести | 139 |
| 9. ЗАПИСИ | 142 |
| 10. ФАЙЛИ | 145 |
| 11. ПРОЦЕДУРИ КЕРУВАННЯ ЗВУКОВИМИ СИГНАЛАМИ | 152 |
| 12. СЛОВНИК | 154 |
| 13. СПИСОК ЛІТЕРАТУРИ | 156 |