

## Оборнені задачі будови і вивчення алгоритмів

### Пряма задача вивчення програмування

Початкове навчання, загальна схема процесу програмування показані на рисунку 1.

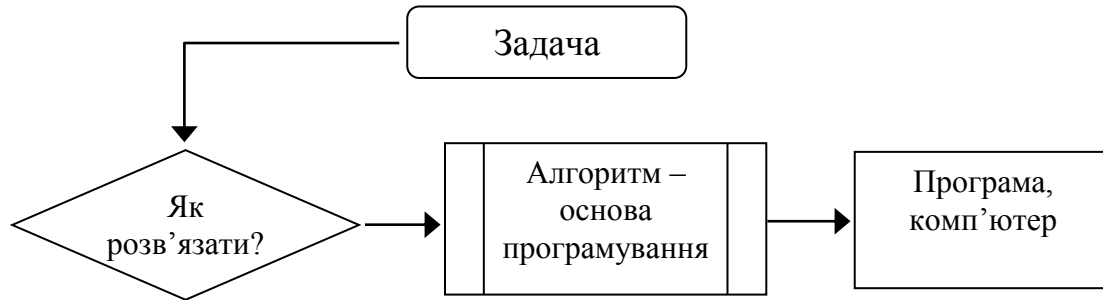


Рис.1. Кроки процесу розробки програми

*Приклад задачі.* Маємо три фігури кругової форми різних радіусів, як на рисунку 2.

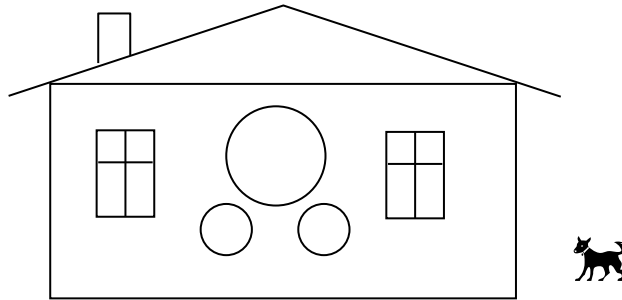


Рис.2.

Більше коло має радіус  $R_v$ , а обидва менших – радіуси  $r_m$ .

Потрібно зафарбувати кола, маючи в наявності фарбу для максимальної загальної площі можливого зафарбування  $P$ .

Очевидно, що треба обрати певний план можливого фарбування залежно від кількості фарби.

- 1) зафарбувати все – коло  $R_v$  і обидва кола  $r_m$ ;
- 2) якщо не вистарчає фарби для п.1, то фарбувати обидва менші кола  $r_m$ ;
- 3) якщо не вистарчає фарби для п.2, то фарбувати одне велике коло  $R_v$ ;
- 4) якщо й тепер не вистарчає фарби – то нічого не фарбувати.

*Алгоритм розв'язку задачі*

- 1) дізнатись (прочитати) конкретні значення  $R_v$ ,  $r_m$ ,  $P$ ;
- 2) обчислити площі великого кола і малих кіл;
- 3) порівняти  $S_v + S_m + S_m$  з  $P$ ;  
якщо вистарчає – то план 1, якщо ні – думаємо далі;
- 4) порівняти  $S_m + S_m$  з  $P$ ;  
якщо вистарчає – то план 2, якщо ні – думаємо далі;
- 5) порівняти  $S_v$  з  $P$ ;  
якщо вистарчає – то план 3, якщо ні – то план 4;
- 6) друкувати результати.

### Реалізація алгоритма мовою Python

1) дізнатись (прочитати) конкретні значення  $R_v$ ,  $r_m$ ,  $P$

```
Rv = float(input('Радіус великого кола? '))  
rm = float(input('Радіус малого кола? '))  
P = float(input('Скільки є фарби? '))
```

2) обчислити площі великого кола і малих кіл

```
Sv = math.pi * math.pow(Rv, 2)  
Sm = math.pi * math.pow(rm, 2)
```

3), 4), 5)

```
if P >= Sv+Sm+Sm :  
    plan=1  
elif P >= Sm+Sm :  
    plan=2  
elif P >= Sv :  
    plan=3  
else:  
    plan=4
```

6) друкувати результати

```
print('Маємо фарби ', P)  
if plan <= 3 :  
    print('Можна реалізувати план ', plan)  
else:  
    print('Для фарбування недостатньо фарби')
```

### Постановка оберненої задачі навчання

Розробка алгоритмів в сучасних комп'ютерних системах має багато методів і технологій. Майбутній розробник має бути добре знайомий з типовими прийомами програмування, а також знати методи і технології. Одним з ефективних методів навчання основ алгоритмізації є метод обернених задач. За таким методом учневі спочатку пропонують готові розв'язки деяких задач. Учень має виконати аналіз розв'язку і спробувати відновити умову задачі, до якої поданий розв'язок, або виконати вручну обчислення за операторами – *обернена* задача навчання. Наступним кроком учневі пропонують вже саму умову подібної задачі, а потрібно записати алгоритм і програму її розв'язування – *пряма* задача навчання. Отже, схема навчання за методом “аналіз-синтез” може виглядати, як на рисунку 3. Кроки 1-2-3-4 повторюємо багатократно для різних типів задач.

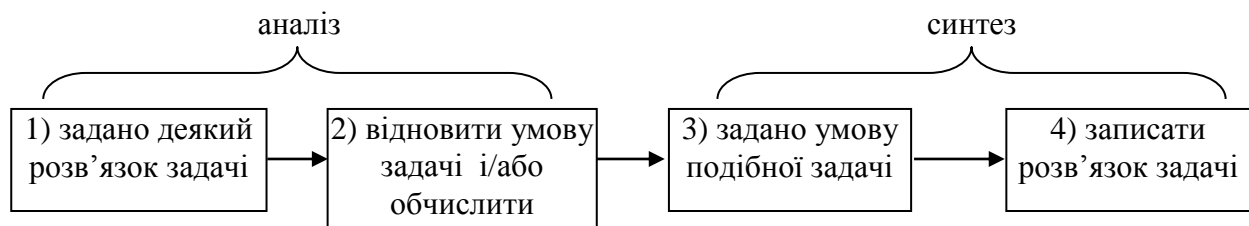


Рис.3. Схема навчання “аналіз-синтез”

## Початкове навчання. Обернені задачі

### Приклади задач аналізу

1. (Обчислення) Яке значення отримає величина  $y$  в результаті виконання таких операторів:

```
int a = 2, b = 3, y;  
y = a++ + --b; y += (++ a - b ++);  
( Відповідь: 6 )
```

2. (Обчислення) Маємо такі оператори C++-програми:

```
int k=8; int b=4; int * p = &k; k = *p + 3*b - k;
```

Яке значення отримає величина  $k$  в результаті виконання цих операторів?

( Відповідь: 12 )

3. (Обчислення) Що надрукує програма

```
#include <iostream>  
int main() {  
    int n; std::cin >> n; int s=0;  
    while (n>0) { s = s*10 + n%10; n /= 10; }  
    std::cout << s; return 0;  
}
```

якщо у вхідному потоці задано '402352' ?

( Відповідь: 253204 )

4. (Відновлення умови) Задано масив 10 дійсних чисел і інші величини:

```
double mas[10], t; int p;
```

Яку задачу розв'язують такі оператори програми:

```
t=0; for (p=0; p<10; p++) if(mas[p]<0) { t+=mas[p]; }
```

( Відповідь: сума від'ємних елементів масиву )

5. (Обчислення з вибором відповіді) Які логічні вирази визначають результат обчислення величини  $z$  як true:

- 1) `int a=1, b=2, c=3; bool z=false; z = !z && (a-1);`
- 2) `int a=1, b=2, c=3; bool z=false; z = z || (a-1);`
- 3) `int a=1, b=2, c=3; bool z=false; z = (a+b)>c && z;`
- 4) `int a=4, b=5, c=6; bool z=false; z = (a+b)<0 || !z && z ;`
- 5) `int a=4, b=5, c=6; bool z=false; z = !!z || !z && c;`

(Відповідь: 5 )

### Приклади задач синтезу

(Задачі синтезу розв'язують після задач аналізу.)

*Скласти програму відповідно до умови задачі.*

1. Від початку доби минуло  $k$  секунд. Який це час в годинах і хвиликах, і в якій половині доби?
2. Задано 5-цифрове ціле число. Порахувати суму цифр, які мають парне значення.
3. Для заданого масиву 14 дійсних чисел обчислити найменше значення серед додатніх елементів.

## Фахове навчання. Обернені задачі на обчислення

### 1. Що надрукує програма

```
#include <iostream>
#include <string>
int main() {
    int n; std::cin >> n; std::string s;
    while (n>0) {
        if (n%2) s = '1' + s; else s = '0' + s;
        n >>= 1;
    }
    std::cout << s; return 0;
}
```

якщо у вхідному потоці задано '39' ?

( Відповідь: 100111 )

Це є задача обчислень арифметичних (%,+), бітових (>>=) і літерних конкатенацій (+).

### 2. Що надрукує програма

```
#include <iostream>
void A(int x, int y) { y = x+1; --x; }
void B(int& x, int y) { y = x+1; --x; }
void C(int& x, int& y) { y = x+1; --x; }
int main() { int x; int y;
    std::cin >> x >> y; A(x,y*y); std::cout << x+y;
    std::cin >> x >> y; B(x,y*y); std::cout << x+y;
    std::cin >> x >> y; B(x,y); std::cout << x << y;
    std::cin >> x >> y; C(x,y); std::cout << x << y;
    return 0;
}
```

якщо у вхідному потоці задано '2 3 2 3 1 5 1 5' ?

( Відповідь: 540502 )

Задача на розуміння визначень функцій і способів передавання параметрів при виклику функцій (за значенням, за адресою, за посиланням), а також розуміння різниці між префіксною і постфіксною формою операцій інкременту і декременту.

3. Рекурсивна функція GetAny() читає цілі числа, надруковані по одному на клавіатурі, і виконує деякі обчислення. Послідовність надрукованих чисел закінчується нулем.

```
int GetAny()
{
    int numb;
    cout << "Print int number: "; cin >> numb;
    if(numb==0) return 0;
    else if(numb<0) return numb+GetAny();
    else return GetAny();
}
```

Який результат res буде обчислено за викликом функції

```
int res=GetAny();
```

при такій послідовності надрукованих чисел: 9, 3, -12, 5, -2, -5, -1, 40, 4, 0.

( Відповідь: -20 )

Задача вимагає розуміння правил будови і виконання рекурсивних функцій.

**4. Складена деяка функція FF():**

```
int FF(int a, int b)
{ int res = 4;
  try
  {
    if(a<0 || b<0) throw '<';
    if(a==b) { throw 100; throw 200; }
    res = a / b;
  }
  catch (int p) { res = res+a+p; }
  catch (char) { res = a=b; }
  catch (...) { b=5; a=10; }
  return res+a;
}
```

Записаний оператор, який викликає функцію FF:

```
int y = FF(8,2) + FF(0,-4);
```

Яке значення отримає величина y в результаті виконання такого оператора?

( Відповідь: 4 )

Задача на знання технологій будови програм (захищені блоки, виняткові ситуації) і правил виконання програм за винятковими ситуаціями. Потрібна максимальна уважність для виконання "ручних" обчислень.

### Фахове навчання. Обернені задачі на відновлення умови

**1. Задано масив 10 цілих чисел і інші величини:**

```
double mas[10], mm; int p;
```

Яку задачу розв'язують такі оператори програми:

а) mm=1; for (p=0; p<10; p++) if(mas[p]>0) { mm=mm\*mas[p]; }

б) mm=1; for (p=-1; p<9; p++) if(mas[p+1]>0) { mm\*=mas[p+1]; }

в) mm=1; for (p=9; p> -1; p--) if(mas[p]>0) { mm\*=mas[p]; mas[p]=mas[p]; }

( Відповідь: добуток додатніх значень масиву – кожний варіант операторів)

В цій задачі використали такі способи будови задач: а – прямий розв'язок; б – ускладнений розв'язок зміщенням індекса; в – аналіз даних в оберненому порядку.

**2. Задано масив з 12 літер і інші величини:**

```
char lett[12]; int x;
```

Яку задачу розв'язує такий фрагмент програми:

а) for (x=0; x<12; x++)  
if (lett[x]>='a' && lett[x]<='z') lett[x]='A'+lett[x]-'a';

б) for (x=0; x<12; x++)  
if (lett[x]>='a' && lett[x]<='z') lett[x]=(int)'A'+lett[x]-(int)'a';

в) for (x=0; x<12; x++)  
if (lett[x]>='a' && lett[x]<='z') lett[x]='A'+lett[x]-(short)'a';

( Відповідь: перетворити масив за правилом: малі латинські букви замінити відповідними великими, решту літер залишити без змін. Всі фрагменти розв'язують ту саму задачу. )

Способи будови задач: а – прямий; б – додатковий параметр int перетворення типу; в – додаткове перетворення типу short, але для частини даних.

3. Визначена така структура для координат точок площини:

```
struct para { double x; double y; };
```

Складена функція для опрацювання масиву розміру  $n$  з координатами точок площини:

```
int AxBy(para P[], int n)
{
    int U=n; int k=n-1;
    for(; k>=0; k--) if(P[k].x<=0 || P[k].y<=0) U--;
    return U;
}
```

Яку задачу розв'язує функція?

( Відповідь: обчислення кількості точок першої чверті площини )

Це є варіант розв'язку задачі шляхом не прямого обчислення, а шляхом відкидання даних протилежної умови. Це суттєво ускладнює відновлення головної умови задачі.

### Фахове навчання. Комплексні обернені задачі

1. ( Аналіз, відновлення умови і обчислення )

Для обчислення кореня рівняння  $f(x)=0.01-\sin(x+0.06)+\cos(x/2)=0$  на відрізку  $[a,b]$  з точністю  $\epsilon$  методом поділу відрізка навпіл складена рекурсивна функція Korin:

```
double Korin(double a, double b, double eps)
{
    double x, f, result;
    x = (a + b) / 2;    f = 0.01 - sin(x + 0.06) + cos(x/2);
    if (b - a <= eps)  result = x;
    else
        if ((0.01-sin(a+0.06) + cos(a/2))*f < 0)  result = Korin(a, x, eps);
        else  result = Korin(x, b, eps);
    return result;
}
```

Вважати, що  $\epsilon > 0$ ,  $a < b$ ,  $f(a) \cdot f(b) < 0$  і  $f(x)$  – неперервна і монотонна функція на відрізку  $[a,b]$ .

Скільки разів буде виконана функція Korin() при виклику Korin(0.8, 1.2, 0.001)?

( Відповідь: 10 )

2. ( Аналіз, відновлення умови, обчислення, вибір відповіді )

Задано визначення структури точки на площині:

```
struct Point {
    double x, y;
    Point() { x = 0; y = 0; }
    Point(double a, double b) { x = a; y = b; }
    bool operator<(Point two)
        { return sqrt(pow(x, 2) + pow(y, 2)) <
                sqrt(pow(two.x, 2) + pow(two.y, 2)); }
};
```

Задача. «Скласти функцію, яка в масиві координат точок площини шукає найближчу точку до початку координат.» Серед запропонованих нижче варіантів виберіть усі, що розв'язують поставлену задачу.

```
1. Point Find( Point M[], int size )
{ Point T = M[0];
  for (int i = 0; i < size; i++)
  { if (sqrt(pow(M[i].x, 2) + pow(M[i].y, 2))
        < sqrt(pow(T.x, 2) + pow(T.y, 2)))  T = M[i];  }
  return  T;
}
```

```
2. Point Find( Point M[], int size )
{ Point T = Point(M[0].x, M[0].y);
  Point * P = M;  int i;
  for (i = 0; i < size - 1; i++)
    { if (*P < *(P + 1))  T = *P; }
  return  T;
}

3. Point Find( Point M[], int size )
{
  Point T = Point(0,0);
  for (int i = 1; i < size; i++)  { if (M[i] < T)  T = M[i]; }
  return  T;
}

4. Point Find( Point M[], int size )
{ Point T = Point(M[0].x, M[0].y);  Point * P;  int i;
  for (P = M, i = 0; i < size;  i++)  { if (*P < T) T = *P;  P++; }
  return  T;
}

5. Point Find( Point M[], int size )
{ Point T = M[0];
  for (int i = 1; i < size; i++)  { if (M[i] < T)  T = M[i]; }
  return  T;
}
```

Запишіть номери правильних варіантів.  
( Відповідь: 1, 4, 5 )

### Фахове навчання. Оборнені задачі на обчислення з вибором відповіді

1. Задача. «Оголосити функцію введення (з клавіатури) одновимірного масиву дійсних чисел довільного розміру.» Серед запропонованих нижче варіантів виберіть усі, що розв'язують поставлену задачу.

```
1. void read_1(double a[], int n)
{ for (int i=0; i<n; ++i) std::cin >> a[i]; }

2. void read_2(double * beg, double * end)
{ for (double * x=beg; x!=end; ++x) std::cin >> *x; }

3. void read_3(double* begin, end)
{ for (double* x=begin; x!=end; ++x) std::cin >> *x; }

4. void read_4(double * beg, double * end)
{ double * x=beg; while (x!=end) std::cin >> *x++; }

5. void read_5(double a[], size_t n)
{ for (size_t i=0; i<n; ++i) std::cin >> a[i]; }
```

Запишіть номери правильних варіантів.  
( Відповідь: 1, 2, 4, 5 )

2. У програмі записані структури, які визначають координати точки на площині та її колір:

```
enum PointColor { white, green, blue, yellow };  
const char * pc[4] = { "white", "green", "blue", "yellow" };  
struct ColoredPoint { double x,y; PointColor c;  
    ColoredPoint(double nx,double ny) {x=nx;y=ny;}};
```

Далі визначають і друкують параметри деякої точки. Запишіть номери варіантів, які записані правильно:

- 1) ColoredPoint ttt(2,4); ttt.c=yellow; cout<<ttt.x<<pc[ttt.c]<<endl;
- 2) ColoredPoint ttt(2,4); ttt.c=yellow;  
 cout<<ttt.x<<pc[(int)ttt.c]<<endl;
- 3) ColoredPoint ttt(10,21); ttt.c=3; cout<<pc[3]<<(ttt.y-ttt.x)<<endl;
- 4) ColoredPoint ttt(-3,(int)green); ttt.c=green;  
 cout<<ttt.x+ttt.y<<pc[ttt.c]<<endl;
- 5) ColoredPoint ttt(000,white); ttt.c=green;  
 cout<<pc[white]<<ttt.y+1<<endl;
- 6) ColoredPoint ttt(5,5); ttt.c="blue"; cout<<pc[2]<<ttt.y<<endl;

( Відповідь: 1, 2, 4, 5 )