

РОЗРОБКА ПРОГРАМ З РОЗГАЛУЖЕННЯМ

3.1 Мета роботи

Освоїти прийоми роботи з текстовими редакторами й основними компонентами інтегрованого середовища програмування C++; придбати практичні навички введення–виведення даних, придбати навички у використанні операторів керування; складання і налагодження програм розгалуженням.

3.2 Методичні вказівки щодо організації самостійної роботи студентів

Під час підготовки до роботи повторити типи константні змінних, способи їхнього опису, використання операторів вибору, а також використання інтегрованого середовища програмування на C++ [1-7, конспект лекцій].

До операторів вибору відносять оператор умовного переходу **if** та оператор-перемикач **switch**

3.2.1 Використання оператора умовного переходу

Оператор вибору **if** використовується для вибору одного з двох напрямів подальшого ходу програми.

Оператор умовного переходу має такий формат запису:

if (вираз) оператор 1;

else оператор 2;

Реалізується оператор **if** таким чином: спочатку обчислюється вираз *i*, якщо значення виразу не дорівнює нулю («істина»), виконується оператор 1, в протилежному випадку – оператор 2 і далі управління передається оператору, що є наступним за умовним оператором **if**, наприклад

```
if(a<b) a++;
```

```
else {b=b+4; a++;}
```

У запису оператора **if** друга частина (тобто **else**) може бути відсутня і тоді, якщо вираз приймає значення «неправда», виконується зразу наступний оператор програми, що розташований за умовним. Таку

конструкцію називають «пропуск оператора».

Якщо при дотриманні або недотриманні умови треба виконати декілька інструкцій програми, то ці інструкції слід об'єднати в групу – укласти у фігурні дужки.

За допомогою вкладених одна в іншу декількох інструкцій **if** можна реалізувати множинний вибір. Синтаксис C++ припускає, що у випадку застосування вкладених умовних операторів кожне **else** відповідає найближчому до нього попередньому **if**.

3.2.2 Оператор–перемикач switch

Оператор–перемикач **switch** призначена для вибору одного з декількох можливих напрямів подальшого ходу програми і має таку форму запису:

```
switch(вираз)
{
    case к.в.1: оператор 1; [break;]
    case к.в.2: оператор 2; [break;]
    .....
    case к.в.n: оператор n; [break;]
    [default: оператор n+1;]
},
```

де **switch**, **case**, **default** – службові слова;

break – оператор (необов'язковий) здійснює вихід з оператору **switch**;

вираз – будь-який вираз одного з цілих типів;

к.в.1, ..., к.в.n – константні вирази, які не можуть повторюватися і не можуть містити змінних чи викликів функцій, зазвичай це ціла (**int**) або символічна (**char**) константа;

оператор 1;...; оператор n+1 – будь-які оператори мови C++.

У процесі виконання цього оператора спочатку обчислюється значення **виразу**, потім це значення послідовно зверху донизу порівнюється зі значеннями константних виразів **к.в.1, ..., к.в.n**. У випадку збігу значень **виразу** і одного з цих константних виразів, та якщо наприкінці гілки розгалуження немає оператора **break**, виконуються всі оператори, починаючи з відповідної гілки. За наявності оператора **break** виконується тільки оператор, що знаходиться у відповідній гілці розгалуження, і керування передається оператору, який розташований за

межами оператора **switch**. Якщо значення **виразу** не збігається з жодним із значенням константних виразів, то виконується гілка **default** і здійснюється вихід з оператора **switch**. У випадку, коли в операторі **switch** відсутня гілка **default** і значення не збігається з жодним із значень константних виразів, відбувається вихід з оператора **switch**.

Наприклад:

```
switch(getchar)
{
case 'a': puts("введена буква a"); break;

case 'b': puts("введена буква b"); break;

case 'c': puts("введена буква c"); break;
default: puts("жодна з трьох букв a, b, c не введена");
           break;
}
```

3.3 Порядок виконання роботи та варіанти індивідуальних завдань

Під час виконання роботи необхідно створити та відлагодити програму для розв'язання задачі, номер якої відповідає порядковому номеру студента в журналі академгрупи.