**Cкладання та виконання алгоритмів опрацювання величин числового типу в навчальному середовищі програмування**

**Мета:** навчитися складати та виконувати алгоритми з величинами числового типу в середовищі програмування CodeBlocks з мовою програмування С++.

**Теоретична частина**

**Основні поняття теми**

**Сталі** — *вид даних, величину яких заборонено змінювати протягом виконання програми. Їх описують після зарезервованого слова* const.

**Змінні** — *вид даних, величину яких дозволено змінювати протягом виконання програми. Їх описують, вказавши їхню назву після зарезервованих слів — назв типів.* Наприклад, int k;.

**Властивості змінної:** назва, тип і поточна величина.

**Запис назви змінної чи сталої** *може містити лише літери латиниці, цифри або нижнє підкреслення і має задовольняти такі вимоги:*

* починатися з літери;
* не містити пропусків, ком чи інших непередбачених знаків;
* не збігатися з зарезервованими словами чи назвами операторами.

**Тип даних** *характеризує допустимі величини для цих даних і сукупністю операцій над ними.*

Числові дані можуть бути двох типів: цілі та дійсні.

**Цілі типи**

|  |  |  |
| --- | --- | --- |
| **Назва типу** | **Обсяг, байтів** | **Діапазон** |
| int | 2 або 4 | –32768..32767 або –2147483648..2147483647 |
| short int | 2 | –32768..32767 |
| unsigned short int | 2 або 4 | 0..65536 або 0..4294967295 |
| long int | 4 | –2147483648..2147483647 |
| unsigned long int | 4 | 0..4294967295 |

Об'єм пам'яті, виділений під величини типу int та unsigned short int, залежать від розрядності операційної системи: 2 байти в 32-розрядній і 4 байти в 64-розрядній системі.

**Дійсні типи**

|  |  |  |
| --- | --- | --- |
| **Назва типу** | **Обсяг, байтів** | **Діапазон** |
| float | 4 | 3.4·10–38..3.4·10+38 |
| double | 8 | 1.7·10–308..1.7·10+308 |
| long double | 10 | 1.18·10–4932..1.18·10+4932 |

Для дійсних чисел вказано діапазони зміни абсолютнної величини (модуля). Дійсні числа можна записувати у форматі з фіксованою крапкою (2.56, –0.3) або у науковому форматі (з рухомою крапкою: –0.23e3=–230, 1.45e–2=0.0145)

**Логічний тип bool.** Змінні цього типу можуть набувати лише величин: false (хибність) і true (істина). Вони займають 1 байт у пам'яті.

**Символьний тип char.** Змінні цього типу можуть набувати лише величин з кодової таблиці комп'ютера ASCII.

**Символьна стала**— *це один символ, узятий в одинарні лапки, або число у 8-, 10- чи 16-й системі числення, яке є кодом символу у таблиці ASCII.* Наприклад,  
char s='f'; d=102;  
Зауважимо: 102 — це десятковий код символу «f». Після такого опису:  
char s='f'; int n=s;  
змінна *n* набуде величини 102.

**Вказівка надання значення** має такий вигляд:  
<*назва*> = <*вираз*>;  
або  
<*назва*1> = <*назва*2> = … = <*назваN*> = <*вираз*>;  
і діє так: обчислюють вираз і його величину надають змінній або декільком змінним. Вираз може містити числа (сталі чи змінні) та назви функцій, з'єднані символами операцій. Змінна і вираз не обов'язково мають бути одного типу. У С++ реалізовано автоматичне перетворення (*узгодження*) типів. Розрізняють *явне* і *неявне* узгодження типів. При перетворенні дійсного числа в ціле відкидають дробову частину (це не є округленням!). Наприклад, після виконання такого коду:

int b,a; float c; a=2; c=4.9; b=a\*c;

змінна b набуде величини 9 після *неявного* перетворення дійсного числа 9.8 у ціле.

**Явне перетворення типів** *задають у круглих дужках перед змінною чи виразом.*

Явне перетворення типів для поданого вище прикладу опису змінних:

|  |  |
| --- | --- |
| **Вказівка** | **Результат виконання** |
| b = (int)  c \* a; | b = (int) 4.9 \* 2 = 4 \* 2 = 8 |
| b = (int) (c \* a); | b = (int) (4.9 \* 2 ) = (int) 9.8 = 9 |

Вираз може містити дані різних числових типів (*змішані* вирази). *Узгодження типів у виразах відбувається з урахуванням пріоритету типів:*

* *дійсні числа мають вищий пріоритет, ніж цілі;*
* *для цілих/дійсних типів вищий пріоритет там, де ширший діапазон зміни;*
* *результат перетворюють до типу з вищим проіоритетом.*

**Оператор введення** (з клавіатури) має такий синтаксис:  
cin >>*змінна*>>*змінна*;

**Оператор виведення** (на екран) має такий синтаксис:

cout << *вираз чи змінна*<< *вираз чи змінна*;

**Cкладний оператор** — це послідовність операторів, відділених один від одного крапкою з комою, що починається з відкритої фігурної дужки { і закінчується закритою фігурною дужкою }:  
{ *оператор*1; *оператор*2; … *операторn*; }

Компілятор сприйме складений оператор як один оператор.

**Умовний оператор** призначено для того, щоб задати вибір тієї чи іншої дії залежно від справдження певної умови. Цей оператор має такий синтаксис:

if (<*умова*>) <*оператор*1>; else <*оператор*2>;

Як умову використовують значення логічного виразу, як оператори — довільний простий або складений оператор, у тому числі і сам умовний оператор. Якщо значення логічного виразу true, то буде виконано *оператор*1, якщо false, то буде виконано *оператор*2.

Якщо згідно з алгоритмом потрібно виконати декілька операторів, тоді потрібно використовувати складений оператор.

**Умовний оператор Switch-Case** (оператор вибору) використовують у тих випадках, коли залежно від значень якої-небудь змінної треба виконати ті чи інші оператори. Він має такий синтаксис:  
switch (<*вираз*>)  
{  
  case <*значення*1> : <*оператор*1>; break;  
  case <*значення*2> : <*оператор*2>; break;  
  …  
  case <*значенняn*> : <*оператор*n>; break;  
  default: <*операторn*+ 1>  
};

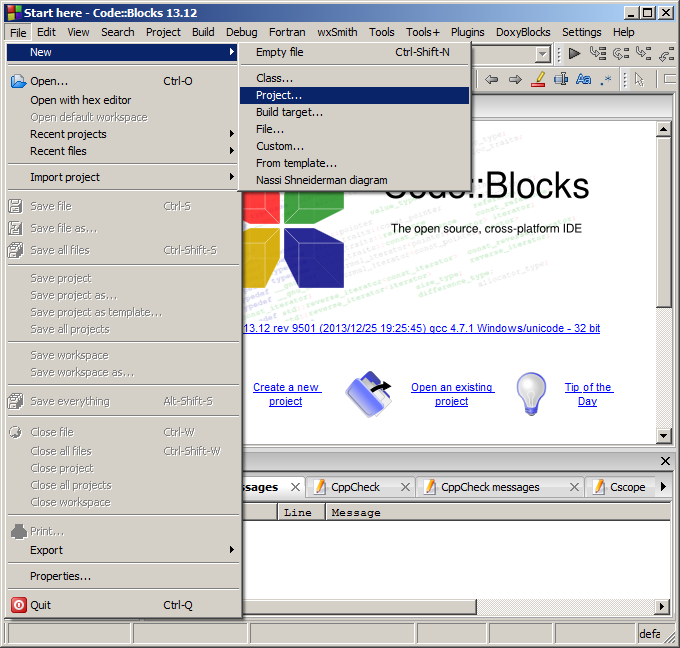
Тут *операторn*+ 1 буде виконано лише у тому випадку, коли значення *виразу* не збігається ні з одним зі *значенняj*. Після кожного логічно відокремленого блоку потрібно використовувати оператор break для виходу з оператора вибору.

**Практична частина**

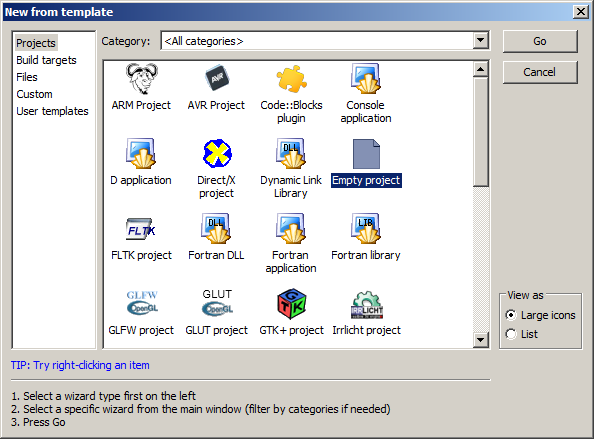
**1. Встановити програму CоdeBlocks та створити новий проект згідно інструкції:** https://www.youtube.com/watch?v=iZrtPgMEK8M.

**2. Алгоритм створення нового проекту**  
(альтернативний метод, якщо не вийшло встановити за відеоінструкцією)

1. Використати вказівку меню *File / New / Progect…*



1. У вікні діалогу *New from template* вибрати *Empty project* і натиснути кнопку *Go*.

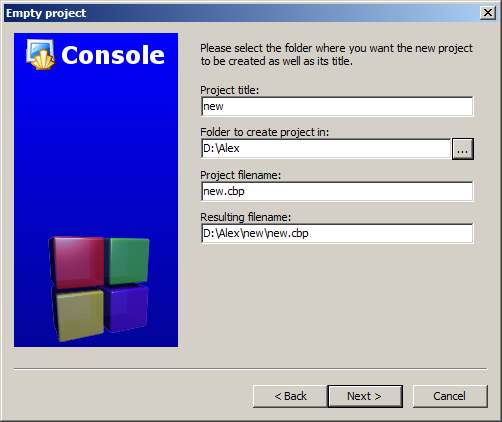


1. У вікні діалогу *Empty project* натиснути кнопку *Next*.

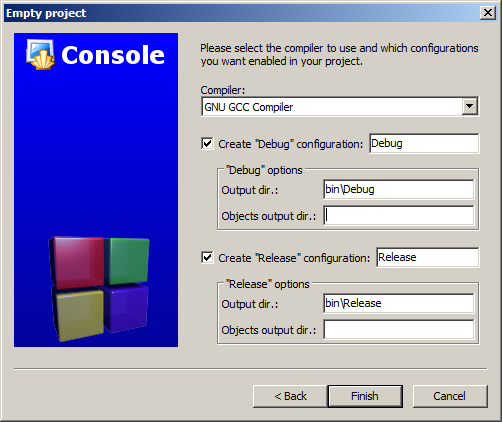


1. У вікні діалогу *Empty project* вказати:
   * у полі *Project title* назву проекта;
   * у полі *Folder to create project in:* теку проекта;
   * у полі *Project filename* назву файла програми

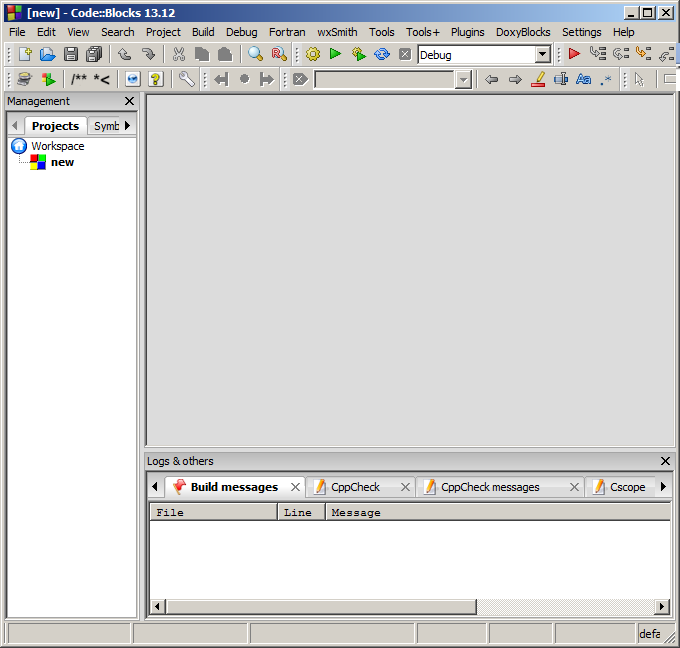
і знову натиснути кнопку *Next*.



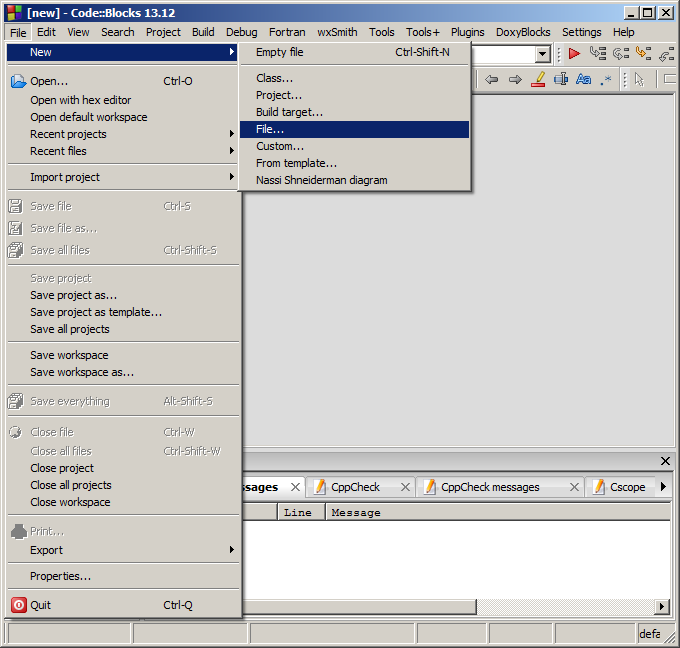
1. У вікні діалогу *Empty project:* у разі потреби змінити налаштування компілятора (наразі це непотрібно робити), натиснути кнопку *Finish*



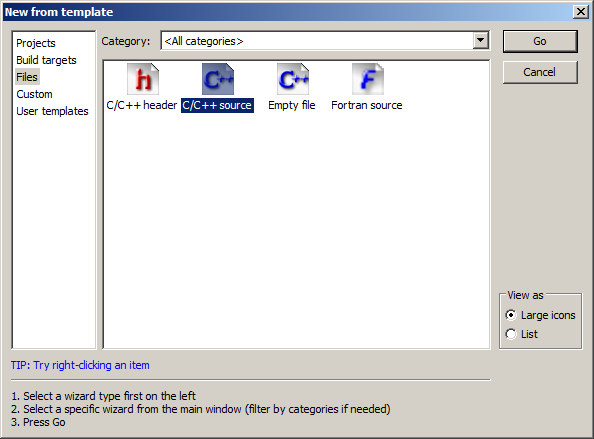
і отримати вікно з назвою проекту.



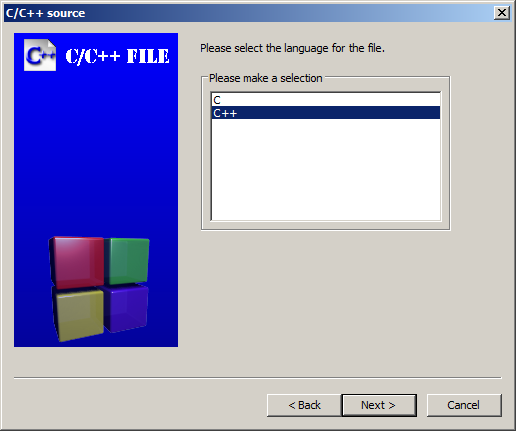
1. Долучити до проекту файл:
   * використати вказівку меню *File / New / File…*;



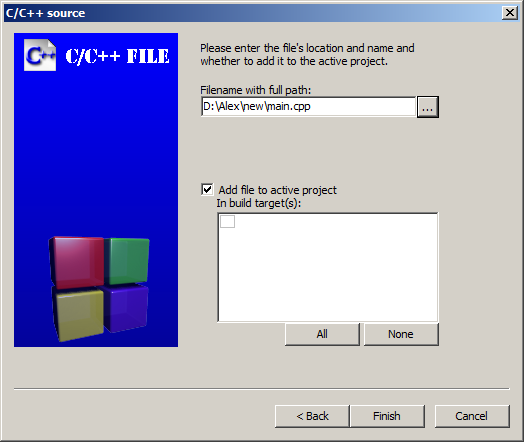
* + у вікні діалогу *New from template* вибрати *C/C++ source* і натиснути кнопку *Go*;



* + у вікні діалогу *C/C++ source* вибрати *C++* і натиснути кнопку *Next*;



* + у вікні діалогу *C/C++ source*, натиснувши кнопку …, вказати у полі *Filename with full path:* шлях до файлу і натиснути кнопку *Finish*.



Не вдаючись у подробиці, подамо приклад програми, що виводить кирилицю.

#include <iostream>

#include <clocale>

using namespace std;

int main()

{

**setlocale(LC\_CTYPE,"rus");**

cout << "Перша проба пера\n";

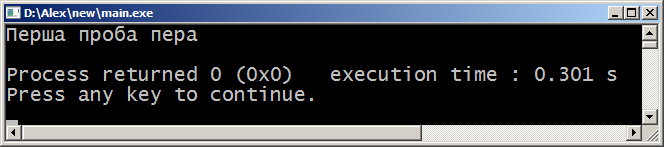
return 0;

}

**Зверніть увагу, що кирилицю можна підключати різними способами: або так, як вказано в прикладі, або setlocale (0, “”).**

Щоб пересвідчитися у цьому, потрібно:

* створити чи відкрити файл програми;
* внести туди поданий вище код;
* використавши наявні на панелі *Compiler* кнопки:  
  http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/77b/11.png *Build* — Побудувати;  
  http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/77b/12.png *Run* — Запустити;  
  http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/77b/13.png *Build and run* — Побудувати і запустити,  
  запустити програму на виконання і побачити таке вікно.



**3. Вироблення практичних навичок**

**Примітка.** Кожне завдання називати Прізвище\_номер завдання відповідно до його порядкового номеру.

**Приклад розв’язання**

**Завдання**. Дано сторони прямокутника **a** та **b**. Знайти його площу **S = a·b** та периметр **P = 2·(a + b)**. Скласти алгоритм рішення та написати програму.

**Рішення**

#include <iostream>

using namespace std;

int main()

{

setlocale (0, “”);

int a, b;

cout << "Введите два числа" << endl;

cin >> a >> b;

int S, P;

S = a \* b;

P = 2 \* (a + b);

cout << "S = " << S << endl;

cout << "P = " << P << endl;

return 0;

}

**Завдання 1.** Написати програму для знаходження середнього арифметичного (**Avr**) двох довільних чисел **a** та **b**. Avr = (a+b)/2.

**Завдання 2.** Дана площа S кола. Знайти його діаметр D та довжину L окружності, яка обмежує це коло, враховуючи, що L = 2·π·R, S = π·R2. В якості значення π використовувати 3.14.

**Завдання 3.** Написати програму, яка запитує курс долара **A** и суму в гривнях **B**. Суму вказує в доларах **S** (S = B/A).

**Відповіді надсилати на електронну пошту licey\_97@ukr.net до 06.04.2020 р. включно у вигляді файлів CodeBlocks або файлу Word з нумерацією завдання та прописаним кодом у вікні CodeBlocks.**