

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА  
Факультет прикладної математики та інформатики  
Кафедра програмування



**ЛАБОРАТОРНА РОБОТА № 9**

Виконала:  
студентка групи ПМОм-11  
Кравець Ольга

Львів 2025

## Семантичні моделі вивчення програмування

### Завдання 1

Запишіть за допомогою розширеної форми Бекуса-Наура синтаксичні визначення таких елементів мови C++:

- текстовий рядок;
- вектор (масив);
- матриця.

*Текстовий рядок:*

string\_initialization ::= "string" variable\_name [ "=" string\_value ] ";"

string\_value ::= "" characters ""

characters ::= character+

character ::= letter | digit | punctuation | space

*Вектор (масив):*

array\_initialization ::= type variable\_name "[" array\_size "]" ";"

type ::= built\_in\_type | user\_type

built\_in\_type ::= "int" | "long" | "float" | "double" | "char"

user\_type ::= identifier

array\_size ::= number

number ::= digit+

digit ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

*Матриця:*

matrix\_initialization ::= type variable\_name "[" array\_size "]" "["  
array\_size "]" ";"

type ::= built\_in\_type | user\_type

built\_in\_type ::= "int" | "long" | "float" | "double" | "char"

user\_type ::= identifier

array\_size ::= number | identifier  
number ::= digit+  
digit ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"  
variable\_name ::= letter ( letter | digit | "\_" ) \*  
identifier ::= variable\_name

## Завдання 2

Запишіть за допомогою розширеної форми Бекуса-Наура синтаксичні визначення таких елементів мови C++:

- визначення структури struct;
- заголовок визначення функції;
- визначення цілої функції;
- визначення блоків опрацювання виключних ситуацій у мові C++ (try, catch).

*Визначення структури struct:*

struct\_initialization ::= "struct" struct\_name "{" [ struct\_body ] "}" ";"  
struct\_name ::= identifier  
struct\_body ::= variable\_declaration\*  
variable\_declaration ::= type variable\_name ";"

*Заголовок визначення функції:*

function\_header ::= return\_type function\_name "(" parameter\_list? ")"  
return\_type ::= "void" | type  
type ::= built\_in\_type | user\_type  
built\_in\_type ::= "int" | "long" | "float" | "double" | "char"  
user\_type ::= identifier  
function\_name ::= identifier  
parameter\_list ::= parameter ( "," parameter ) \*

parameter ::= type parameter\_name  
parameter\_name ::= identifier

*Визначення цілої функції:*

function\_definition ::= function\_header "{" function\_body "}"  
function\_body ::= statement\* return\_statement  
statement ::= variable\_declaration | expression\_statement |  
control\_structure  
return\_statement ::= "return" expression ";"

*Визначення блоків опрацювання виключних ситуацій у мові C++ (try, catch):*

try\_catch\_block ::= try\_block catch\_block+  
try\_block ::= "try" "{" try\_body "}"  
try\_body ::= statement\*  
catch\_block ::= "catch" "(" exception\_parameter ")" "{" catch\_body "}"  
exception\_parameter ::= type identifier  
catch\_body ::= statement\*

statement ::= variable\_declaration | expression\_statement |  
control\_structure | throw\_statement  
throw\_statement ::= "throw" expression ";"  
expression\_statement ::= expression ";"  
expression ::= identifier | literal | function\_call | expression operator  
expression

### **Завдання 3**

Запишіть свій варіант моделі денотативної семантики навчання для кожного з перелічених вище в завданнях 1 і 2 елементів мови

C++. Моделі записати в формі алгебри виду  $UA_i = \langle ti; \{opt_i\} \rangle$  або виду  $UA_i = \langle \{par\}_i; \{operators_i: read, write\} \rangle$ .

#### *Текстовий рядок*

$UA\_string = \langle string; \{=, +, [], length, substr, find\} \rangle$

#### *Вектор (масив)*

$UA\_array = \langle array<T>; \{=, [], size, at, push\_back, pop\_back\} \rangle$

#### *Матриця*

$UA\_matrix = \langle matrix<T>; \{=, index(i,j), rows, cols, transpose, get\_row, get\_col\} \rangle$

#### *Визначення структури (struct)*

$UA\_struct = \langle \{field1:type1, field2:type2, \dots\}; \{read\_field, write\_field, assign\_struct\} \rangle$

#### *Заголовок функції*

$UA\_func\_header = \langle \{return\_type, name, parameters\}; \{define\_signature, check\_types\} \rangle$

#### *Повне визначення функції*

$UA\_func = \langle \{header, body\}; \{read\_inputs, execute\_body, return\_output\} \rangle$

#### *Блоки try-catch*

$UA\_try\_catch = \langle \{try\_block, catch\_blocks\}; \{execute\_try, handle\_exception\} \rangle$

## Завдання 4

Для кожного елемента мови C++, перерахованого в завданнях 1 і 2, складіть умови задач на програмування для учнів, які б розкривали семантику відповідного елемента (по одній задачі на кожний елемент), а не просто отримання результату. Записати лише самі умови задач, програми писати не треба.

### *Рядок*

Скласти програму, яка зчитує рядок з клавіатури та перевіряє, чи міститься у ньому підрядок "abc". Якщо так - вивести повідомлення про знайдений підрядок.

### *Вектор*

Написати програму, яка зчитує набір цілих чисел у вектор, поки не буде введено 0. Після цього обчислити та вивести середнє арифметичне всіх елементів вектора.

### *Матриця*

Задано матрицю розміром 3x3. Необхідно транспонувати її (поміняти місцями рядки та стовпці) та вивести результат.

### *Struct*

Описати структуру Student з полями: ПІБ, вік, середній бал. Створити масив з кількох студентів і вивести на екран їх дані.

### *Заголовок функції*

Написати заголовок функції sum, яка приймає два параметри типу float і повертає їх суму. Реалізовувати тіло функції не потрібно - важливо правильно оголосити типи.

### *Функція*

Скласти функцію `multiply`, яка приймає два параметри типу `int`, обчислює їх добуток та повертає результат. Викликати функцію з головної програми.

### *Try/catch*

Написати програму, яка виконує ділення двох чисел, зчитаних з клавіатури. Якщо знаменник дорівнює нулю, програма повинна перехопити виняток і вивести повідомлення про помилку.