Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет комп’ютерних систем та автоматики

Кафедра КСУ

Лабораторна робота №12

з дисципліни: «Компютерні технології та програмування»

Тема: «Постачальник – споживач»

Виконав

студент групи 2АКІТ-20б

Н. Бондар

Перевірив

к.т.н., доц.кафедри КСУ

Севастьянов В. М.

м.Вінниця, ВНТУ 2021

**Лабораторна робота №12**

**Тема: «Постачальник – споживач»**

**Варіант 1.**

**Завдання :**

Порядок виконання і звітування

1. Створити програму на мові С++ згідно варіанту.

2. Відкомпілювати та відлагодити програму.

3. Розробити набір тестів і перевірити роботу програми на них.

4. Відповісти на контрольні запитання.

5. До кожної програми скласти схему програми.

6. Зробити висновки.

Звіт по лабораторній роботі має складатися з титульної сторінки,

лістингів програм, схем програм, висновків по роботі.

**Варіант 1. Задача про Вінні-Пуха або правильні бджоли.**

В одному лісі живуть n бджіл і один ведмідь, які використовують один

горщик меду, місткістю Н ковтків. Спочатку горщик порожній. Поки горщик

не наповниться, ведмідь спить. Як тільки горщик заповнюється, ведмідь

прокидається і з'їдає весь мед, після чого знову засинає. Кожна бджола

багаторазово збирає по одному ковтку меду і кладе його в горщик. Бджола,

яка приносить останню порцію меду, будить ведмедя. Створити

багатопотоковий додаток, що моделює поведінку бджіл і ведмедя.

**Хід роботи :**

**1. Ретельно вивчаю поставлене передімною завдання.**

**2. Створюю програму на мові С згідно варіанту використавши середовище програмування Dev-C++ 4.0:**

**3. Відкомпільовую та відлагоджую програму.**

**4. Розробляю набір тестів і перевіряю роботу програми на них.**

**5. Відповідаю на контрольні запитання.**

**6. Роблю висновки.**

**0. Обявляю необхідні змінні :**

int honey\_pot = 0;

int honey\_pot\_size = 5;

**1. Розписую функцію eat\_all\_homey :**

int eat\_all\_homey(int honey\_pot) {

honey\_pot = 0;

std::cout << "\n Bear has eaten all honey\n";

sleep(2);

std::cout << "\n Honey pot is empty : " << honey\_pot << '\n';

sleep(2);

std::cout << "\nRun everything again\n\n";

sleep(2);

return honey\_pot;

}

**2. Розписую функцію put\_some\_homey :**

int put\_some\_homey(int honey\_pot) {

if (honey\_pot >= honey\_pot\_size) {

std::cout << "\nBee gonna wake up the Bear\n";

sleep(2);

std::cout << "\n Bear woke up\n";

sleep(2);

auto tbear = std::async(eat\_all\_homey, honey\_pot);

honey\_pot = tbear.get();

}

std::cout << "Bee has put some honey\n Honey pot : " << ++honey\_pot << '\n';

return honey\_pot;

}

**3. Розписуюю головну функцію main :**

int main()

{

while (true) {

auto tbee = std::async(put\_some\_homey, honey\_pot);

honey\_pot = tbee.get();

sleep(2);

}

return 0;

}

**Блок-Схема :**

ПОЧАТОК

|

Поки вірно <------------------------\

|Ні |Так |

| Створити потік бджола |

| | |

| Якщо горщик повний |

| |Ні |Так |

| | Вивести відповідний текст |

| | | |

| | Запустити потік ведмедя |

| | | |

| | Отримати стан горщика |

| \------>| |

| | |

| Вивести вміст горщика |

| | |

| Отримати стан горщика >---------/

\-->|

КІНЕЦЬ

**Лістинг коду :**

/\*

Варіант 1. Задача про Вінні-Пуха або правильні бджоли.

В одному лісі живуть n бджіл і один ведмідь, які використовують один

горщик меду, місткістю Н ковтків. Спочатку горщик порожній. Поки горщик

не наповниться, ведмідь спить. Як тільки горщик заповнюється, ведмідь

прокидається і з'їдає весь мед, після чого знову засинає. Кожна бджола

багаторазово збирає по одному ковтку меду і кладе його в горщик. Бджола,

яка приносить останню порцію меду, будить ведмедя. Створити

багатопотоковий додаток, що моделює поведінку бджіл і ведмедя.

\*/

#include<iostream>

#include<thread>

#include<future>

#include<unistd.h>

int honey\_pot = 0;

int honey\_pot\_size = 5;

/\*

Блок-Схема :

ПОЧАТОК

|

Поки вірно <------------------------\

|Ні |Так |

| Створити потік бджола |

| | |

| Якщо горщик повний |

| |Ні |Так |

| | Вивести відповідний текст |

| | | |

| | Запустити потік ведмедя |

| | | |

| | Отримати стан горщика |

| \------>| |

| | |

| Вивести вміст горщика |

| | |

| Отримати стан горщика >---------/

\-->|

КІНЕЦЬ

\*/

int eat\_all\_homey(int honey\_pot) {

honey\_pot = 0;

std::cout << "\n Bear has eaten all honey\n";

sleep(2);

std::cout << "\n Honey pot is empty : " << honey\_pot << '\n';

sleep(2);

std::cout << "\nRun everything again\n\n";

sleep(2);

return honey\_pot;

}

int put\_some\_homey(int honey\_pot) {

if (honey\_pot >= honey\_pot\_size) {

std::cout << "\nBee gonna wake up the Bear\n";

sleep(2);

std::cout << "\n Bear woke up\n";

sleep(2);

auto tbear = std::async(eat\_all\_homey, honey\_pot);

honey\_pot = tbear.get();

}

std::cout << "Bee has put some honey\n Honey pot : " << ++honey\_pot << '\n';

return honey\_pot;

}

int main()

{

while (true) {

auto tbee = std::async(put\_some\_homey, honey\_pot);

honey\_pot = tbee.get();

sleep(2);

}

return 0;

}

**Висновок :** виконуючи дану лабораторно роботу, я навчився працювати з потоками, та застосовувати їх у ході виконання програми.