Звіт

Автор: Чугунов В.Ю

КІТ-119а

Лабораторна робота №16

**СИСТЕМНА РОБОТИ З ДИНАМІЧНОЮ ПАМ’ЯТТЮ**

**Мета:** дослідити особливості мови С++ при роботі з динамічною пам’яттю.

1 ЗАВДАННЯ ДО РОБОТИ

**Індивідуальне завдання.** Для власного типу перевантажити оператори new / new [] та delete / delete [].

2 РОЗРОБКА АЛГОРИТМІВ ФУНКЦІЙ

**Клас Unit**

**private поля:**

const char \*name – ім’я бойової одиниці

int healPoints – кількість heal point-ів бойової одиниці

**Оператори:**

**void \*operator new ( size\_t size )** Призначення: перевизначення оператора new.  
 Опис роботи: використовується функція malloc для виділення пам’яті.

**void \*operator new[] ( size\_t size )** Призначення: перевизначення оператора new[].  
 Опис роботи: використовується функція malloc для виділення пам’яті.

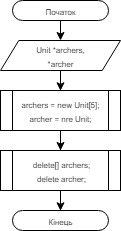
**void operator delete ( void \*p )** Призначення: перевизначення оператора delete.  
 Опис роботи: використовується функція free для звільнення пам’яті.

**void operator delete[] ( void \*p )** Призначення: перевизначення оператора delete[].  
 Опис роботи: використовується функція free для звільнення пам’яті.

**Функції:**

**int main ( )** Призначення: головна функція.  
 Опис роботи:   
 1) виділення пам’яті під масив та під один елемент;  
 2) звільнення пам’яті, виділеної під масив та під один елемент.  
 Схема алгоритму подана на рис. 1.

**Схеми алгоритмів функцій.**

  
Рисунок 1 – Схема алгоритму функції main

3 ТЕКСТ ПРОГРАМИ

**Текст файлу main.cpp**

#include <iostream>

class Unit

{

const char \*name;

int healPoints;

public:

Unit() : name("Archer"), healPoints(100) {}

};

void \*operator new(size\_t size)

{

std::cout << "New\n";

void \*p;

p = malloc(size);

return p;

}

void \*operator new[](size\_t size)

{

std::cout << "New[]\n";

void \*p;

p = malloc(size);

return p;

}

void operator delete(void \*p)

{

std::cout << "Delete\n";

free(p);

}

void operator delete[](void \*p)

{

std::cout << "Delete[]\n";

free(p);

}

int main()

{

Unit \*archers;

archers = new Unit[5];

delete[] archers;

Unit \*archer;

archer = new Unit;

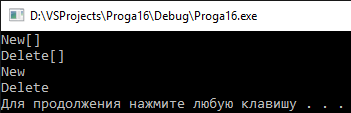
delete archer;

system("pause");

return 0;

}

4 РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ

  
Рисунок 2 – Результат роботи програми

ВИСНОВКИ

При виконанні лабораторної роботи набуто практичних навичок щодо перевизначення та використання операторів new/new[] та delete/delete[].