

Лабораторна робота № 1. Класи

Тема. Класи та специфікатори доступу. Інкапсуляція. Константи.

Мета роботи. Отримати базові знання про класи. Дослідити механізм інкапсуляції.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Малюга Андрій Володимирович,
- НТУ “ХПІ” КІТ 102.8а
- Варіант 13

1.2 Загальне завдання

Для предметної галузі з табл. 1.2 розробити два класи:

☐ клас, що відображає сутність «базового класу». При цьому, в даному класі повинно бути мінімум три числових поля (бажано, щоб одне з цих полів було унікальним ідентифікатором об'єкту);

☐ клас, що має в собі динамічний масив об'єктів базового класу та має в собі методи додавання, видалення елемента, отримання елемента по індексу (або ідентифікатору), вивід усіх елементів на екран. Рекомендовані сигнатури методів:

☐ додавання:

`void CList::addPhone(PHONE& phone);`

☐ видалення:

`void CList::removePhone(int index);`

☐ отримання по індексу:

`CPHONE& CList::getPhone(int index);`

☐ вивід усіх елементів:

`void CList::showAll();` при цьому цей метод повинен викликати метод `getPhone(index)`, щоб не було дублювання коду.

1.3 Завдання

Прикладна галузь – програмне забезпечення

Базова галузь – програма, що виконується

2 ОПИС ПРОГРАМИ

2.1 Функціональне призначення

За допомогою цієї програми можна створити масив об'єктів, додавати та видаляти об'єкти, виводити вміст масиву на екран та вивід об'єкта по індексу, пошук по імені. Також у цій програмі реалізоване зручне меню спілкування з користувачем та модульні тести, що підтверджують коректність роботи деяких функцій.

У програмі присутні модульні тести, що демонструють коректність роботи методів додавання та видалення об'єктів.

2.2 Опис логічної структури

Діаграма класу зображено на рисунку №1

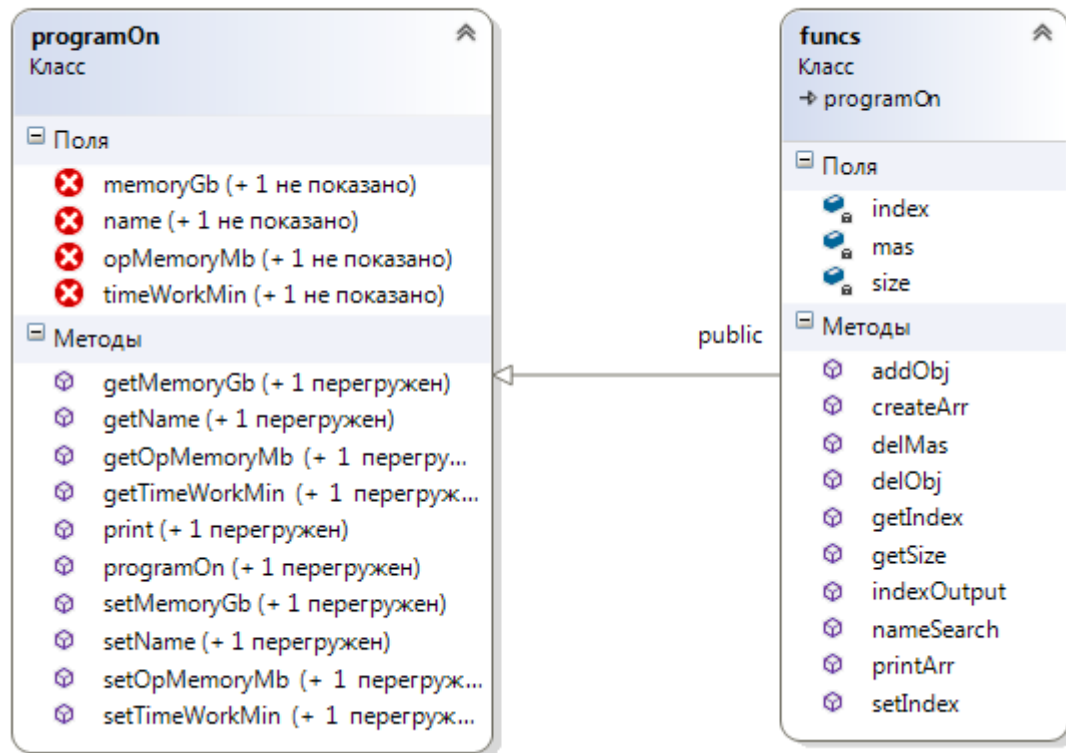


Рисунок №1 – діаграма класу

Метод створення масиву об'єктів зображено на рисунку №2

```

3 void funcs::createArr() {
4     string n;
5     float om, mg, twm;
6
7     cout << "Enter size of array: " << endl;
8     cin >> size;
9     mas = new programOn[size];
10
11     for (int i = 0; i < size; i++) {
12         cout << "Enter name of program:" << endl;
13         cin >> n;
14         cout << "Enter amount of consumed RAM(Mb):" << endl;
15         cin >> om;
16         cout << "Enter occupied amount of hard disk memory(Mg):" << endl;
17         cin >> mg;
18         cout << "Enter time of work (in minutes):" << endl;
19         cin >> twm;
20
21         mas[i].setName(n);
22         mas[i].setOpMemoryMb(om);
23         mas[i].setMemoryGb(mg);
24         mas[i].setTimeworkMin(twm);
25     }
26 }
27

```

Рисунок №2 – метод створення масиву об'єктів

Метод виводу на екран об'єкта зображена на рисунку №3

```

void const programOn::print() {
    cout << "Name of program: " << name << endl;
    cout << "Amount of consumed RAM(Mb): " << opMemoryMb << endl;
    cout << "Occupied amount of hard disk memory(Gb): " << memoryGb << endl;
    cout << "Time of work (in minutes): " << timeworkMin << endl;
}

```

Рисунок №3 – метод виводу об'єкта на екран

Метод додавання зображено на рисунку №4

```

3  void funcs::addObj(int ind, string n, float mg, float om, float twm) {
4      index = ind - 1;
5      programOn *timeMas = new programOn[size];
6      for (int i = 0; i < size; i++) {
7          timeMas[i] = mas[i];
8      }
9      delete[] mas;
10     mas = new programOn[size+1];
11
12     mas[index].setName(n);
13     mas[index].setOpMemoryMb(om);
14     mas[index].setMemoryGb(mg);
15     mas[index].setTimeworkMin(twm);
16
17     for (int i = 0; i < size; i++) {
18         if (i >= index) {
19             mas[i + 1] = timeMas[i];
20         }
21         else {
22             mas[i] = timeMas[i];
23         }
24     }
25     size++;
26
27     delete[] timeMas;
28 }

```

Рисунок №4 – метод додавання об'єкта

Методи виводу масиву об'єктів на екран, отримання та запис індексу зображено на рисунку №5

```

30 void funcs::printArr() {
31     for (int i = 0; i < size; i++) {
32         mas[i].print();
33     }
34 }
35
36 void funcs::setIndex(int ind) {
37     index = ind - 1;
38 }
39
40 int funcs::getIndex(int ind) {
41     ind = index;
42     return ind;
43 }

```

Рисунок №5 - методи виводу масиву об'єктів на екран, отримання та запис індексу

Методи вивода об'єкту по індексу, пошук об'єкта за ім'ям, отримання розміру масиву та видалення масиву зображено на рисунку №6

```

72 void funcs::indexOutput() {
73     mas[index].print();
74 }
75
76 void funcs::nameSearch(string n) {
77     string na;
78     for (int i = 0; i < size; i++) {
79         na = mas[i].getName(na);
80         if (na == n) {
81             mas[i].print();
82         }
83     }
84 }
85
86 int funcs::getSize(int sizeMas) {
87     sizeMas = size;
88     return sizeMas;
89 }
90
91 void funcs::delMas() {
92     delete[] mas;
93 }

```

Рисунок №7 – методи вивода об'єкту по індексу, пошук об'єкта за ім'ям, отримання розміру масиву та видалення масиву

Метод видалення об'єкта зображено на рисунку №8

```

30 void funcs::delobj(int ind) {
31     index = ind - 1;
32     size--;
33     programOn *timeMas = new programOn[size + 1];
34     for (int i = 0; i < size; i++) {
35         if (i < index) {
36             timeMas[i] = mas[i];
37         }
38         else {
39             if (size != 0) {
40                 timeMas[i] = mas[i + 1];
41             }
42         }
43     }
44     delete[] mas;
45
46     mas = new programOn[size];
47
48     for (int i = 0; i < size; i++) {
49         mas[i] = timeMas[i];
50     }
51
52
53     delete[] timeMas;
54 }
55

```

Рисунок №8 – видалення об'єкта

Головна функція main зображена на рисунку №9

```
1  #include "Header.h"
2
3  int main() {
4      funcs ops;
5      menu(ops);
6  }
```

Рисунок №9 – головна функція main

Функція меню спілкування зображена на рисунку №10

```

3 void menu(funcs ops) {
4     int num;
5     int ind;
6     int sizeMas = 0;
7     string n;
8     float om, mg, twm;
9
10    while (true) {
11        if (sizeMas == 0) {
12            cout << "The array is empty. Do you want to create array?" << endl;
13            cout << "(1)Yes" << endl;
14            cout << "(0)No" << endl;
15            cin >> num;
16            if (num == 1) {
17                ops.createArr();
18                sizeMas = ops.getSize(sizeMas);
19                system("cls");
20            }
21            else {
22                exit(0);
23            }
24        }
25        else {
26            cout << "What function do you want to cause the list?" << endl;
27            cout << "(0)Exit from program" << endl;
28            cout << "(1)output array on display" << endl;
29            cout << "(2)name search" << endl;
30            cout << "(3)to delete object from array" << endl;
31            cout << "(4)to add new object to array" << endl;
32            cout << "(5)index output on display" << endl;
33            cin >> num;
34            system("cls");
35            switch (num) {
36                case 1:
37                    ops.printArr();
38                    cout << endl;
39                    system("pause");
40                    system("cls");
41                    break;
42                case 2:
43                    cout << "('.' exit from search)Enter name of program from array: ";
44                    cin >> n;
45                    if (n == "0") {
46                        system("cls");
47                        break;
48                    }
49                    else {
50                        ops.nameSearch(n);
51                        cout << endl;
52                        system("pause");
53                        system("cls");
54                        break;
55                    }
56                case 3:
57                    cout << "Enter index of object for delete it: ";
58                    cin >> ind;
59                    sizeMas = ops.getSize(sizeMas);
60                    if (ind <= sizeMas) {
61                        ops.setIndex(ind);
62                        ops.delObj(ind);
63                        sizeMas = ops.getSize(sizeMas);
64                        system("cls");
65                        break;
66                    }
67                    else {
68                        cout << "There is no object in the array with this index" << endl;
69
70                        system("pause");
71                        system("cls");
72                        break;

```

```

73     }
74     case 4:
75         cout << "Enter index of object for add it to array: ";
76         cin >> ind;
77         sizeMas = ops.getSize(sizeMas);
78         if (ind <= sizeMas + 1 && ind >= 1) {
79             ops.setIndex(ind);
80
81             cout << "Enter name of program:" << endl;
82             cin >> n;
83             cout << "Enter amount of consumed RAM(Mb):" << endl;
84             cin >> om;
85             cout << "Enter occupied amount of hard disk memory(Mg):" << endl;
86             cin >> mg;
87             cout << "Enter time of work (in minutes):" << endl;
88             cin >> twm;
89             ops.addObj(ind, n, mg, om, twm);
90
91             sizeMas = ops.getSize(sizeMas);
92             system("cls");
93             break;
94         }
95         else {
96             cout << "There is no object in the array with this index" << endl;
97
98             system("pause");
99             system("cls");
100            break;
101        }
102        break;
103    case 5:
104        cout << "Enter index of object from array: ";
105        cin >> ind;
106        sizeMas = ops.getSize(sizeMas);
107        if (ind <= sizeMas && ind >= 1) {
108            ops.setIndex(ind);
109            ops.indexOutput();
110
111            system("pause");
112            system("cls");
113            break;
114        }
115        else {
116            cout << "There is no object in the array with this index" << endl;
117
118            system("pause");
119            system("cls");
120            break;
121        }
122    case 0:
123        ops.delMas();
124        exit(0);
125    }
126 }
127 }
128 }

```

Рисунок №10 – функція меню спілкування

Методи setter та getter зображено на рисунку №11


```

4  void programOn::setName(string n) {
5      programOn::name = n;
6  }
7
8  void programOn::setOpMemoryMb(float om) {
9      programOn::opMemoryMb = om;
10 }
11
12 void programOn::setMemoryGb(float mg) {
13     programOn::memoryGb = mg;
14 }
15
16 void programOn::setTimeWorkMin(float twm) {
17     programOn::timeWorkMin = twm;
18 }
19
20
21 //getter
22 string programOn::getName(string n) {
23     n = programOn::name;
24     return n;
25 }
26
27 float programOn::getOpMemoryMb(float om) {
28     om = programOn::opMemoryMb;
29     return om;
30 }
31
32 float programOn::getMemoryGb(float mg) {
33     mg = programOn::memoryGb;
34     return mg;
35 }
36 float programOn::getTimeWorkMin(float twm) {
37     twm = programOn::timeWorkMin;
38     return twm;
39 }

```

Рисунок №11 – методи setter та getter

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма може бути використана для створення масиву об'єктів. Програма має методи додавання, видалення об'єктів, пошук об'єктів по імені та вивід по індексу з масиву. Меню робить роботу з цією програмою зручною.

ВИСНОВКИ

В інтегрованому середовищі Visual Studio розроблена програма мовою C. Засоби налагодження дозволяють за допомогою меню спілкування створити масив об'єктів, змінювати його за допомогою методів класів.