

## Практическое занятие № 16 «Конструкторы и деструкторы класса»

### Учебные цели:

- получение умений и навыков работы с конструкторами и деструкторами классов C++.

### Воспитательные цели:

- формировать диалектико-материалистическое мировоззрение;
- формировать навыки самостоятельности и дисциплинированности;
- стимулировать активную познавательную деятельность обучающихся, способствовать формированию у них творческого мышления.

**Категория слушателей:** 2,3 курс.

**Время:** 90 мин.

**Место проведения:** компьютерный класс.

**Материально-техническое обеспечение:**

1) персональный компьютер *IBM PC* с операционной системой Windows XP; 2) среда разработки приложений *Visual C++.NET*.

### ПЛАН ПРАКТИЧЕСКОГО ЗАНЯТИЯ

| Учебные вопросы                    | Время, мин |
|------------------------------------|------------|
| Вступительная часть .....          | 5          |
| 1. Конструкторы класса в C++. .... | 40         |
| 2. Деструкторы класса в C++ .....  | 40         |
| Заключительная часть .....         | 5          |

### МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОТРАБОТКЕ УЧЕБНЫХ ВОПРОСОВ

**Задание 1.** В соответствии с Вашим вариантом создайте конструктор по умолчанию и деструктор, для заполнения (очистки) полей класса заданной информацией:

1. Описать класс с именем *STUDENT*, содержащий следующие поля:
  - Фамилия,
  - Имя;
  - Отчество;
  - Номер группы;
  - Успеваемость (массив из 5 элементов).

а) Дополнительно введите поля название института и номер курса. С помощью конструктора при инициализации массива объектов класса *STUDENT* установите значения этих полей равными по умолчанию «VGUIT» и 1 соответственно.

б) Создайте деструктор, который обнулит поле номер курса и очистит поле название института, для объектов, чей номер группы равен 23.

Выведите на экран результаты работы программы.

2. Описать класс с именем STUDENT, содержащий следующие поля:

- Фамилия и инициалы,
- Название факультета;
- Название специальности обучения;
- Номер группы;
- Успеваемость (массив из 5 элементов).

а) Дополнительно введите поля название института и номер курса. С помощью конструктора при инициализации массива объектов класса STUDENT установите значения этих полей равными по умолчанию «VGUIT» и 1 соответственно.

б) Создайте деструктор, который обнулит поля номер курса и очистит поле название института, для объектов, чей номер группы равен 23.

Выведите на экран результаты работы программы.

3. Описать класс с именем STUDENT, содержащий следующие поля:

- Фамилия и инициалы,
- Пол;
- Порядковый номер
- Номер группы;
- Успеваемость (массив из 5 элементов).

а) Дополнительно введите поля название института и номер курса. С помощью конструктора при инициализации массива объектов класса STUDENT установите значения этих полей равными по умолчанию «VGUIT» и 1 соответственно.

б) Создайте деструктор, который обнулит поле номер курса и очистит поле название института, для объектов, чей номер группы равен 23.

Выведите на экран результаты работы программы.

4. Описать класс с именем AEROFLOT, содержащий следующие поля:

- Название пункта назначения рейса;
- Номер рейса;
- Тип самолёта;
- Количество свободных мест.

а) Дополнительно введите поля название аэропорта и год основания аэропорта. С помощью конструктора при инициализации массива объектов класса AEROFLOT установите значения этих полей равными по умолчанию «Vnukovo» и 1986 соответственно.

б) Создайте деструктор, который обнулит поле год основания аэропорта и очистит поле название аэропорта, для объектов, чей номер рейса равен 555.

Выведите на экран результаты работы программы.

5. Описать класс с именем AEROFLOT, содержащий следующие поля:

- Название пункта назначения рейса;
- Номер рейса;
- Тип самолёта;
- Стоимость билета.

а) Дополнительно введите поля название аэропорта и год основания аэропорта. С помощью конструктора при инициализации массива объектов класса AEROFLOT установите значения этих полей равными по умолчанию «Vnukovo2» и 1960 соответственно.

б) Создайте деструктор, который обнулит поле год основания аэропорта и очистит поле название аэропорта, для объектов, чей номер рейса равен 777.

Выведите на экрана результаты работы программы.

6. Описать класс с именем SPISOK, содержащий следующие поля:

- Фамилия,
- Имя;
- Отчество;
- Номер телефона;
- Стаж работы;
- День рождения;
- Месяц рождения;
- Год рождения.

а) Дополнительно введите поля название Фирма и номер дома, где расположена Фирма. С помощью конструктора при инициализации массива объектов класса SPISOK установите значения этих полей равными по умолчанию «Sberbank» и 56 соответственно.

б) Создайте деструктор, который обнулит поле номер дома и очистит поле название Фирма, для объектов, чей стаж работы более 25 лет.

Выведите на экрана результаты работы программы.

7. Описать класс с именем SPISOK, содержащий следующие поля:

- Фамилия,
- Имя;
- Отчество;
- Номер телефона;
- Пол;
- День рождения;
- Месяц рождения;
- Год рождения.

а) Дополнительно введите поля название Фирма и номер дома, где расположена Фирма. С помощью конструктора при инициализации массива объектов класса SPISOK установите значения этих полей равными по умолчанию «VTB-24» и 16 соответственно.

б) Создайте деструктор, который обнулит поле номер дома и очистит поле название Фирма, для объектов, чей стаж работы равен 30 лет.

Выведите на экран результаты работы программы.

8. Описать класс с именем SPISOK, содержащий следующие поля:

- Фамилия,
- Имя;
- Отчество;
- Номер телефона;
- Пол;
- Должность;
- Стаж работы.

а) Дополнительно введите поля название Фирма и номер дома, где расположена Фирма. С помощью конструктора при инициализации массива объектов класса SPISOK установите значения этих полей равными по умолчанию «SeverStal» и 32 соответственно.

б) Создайте деструктор, который обнулит поле номер дома и очистит поле название Фирма, для объектов, чей стаж службы меньше 20 лет.

Выведите на экран результаты работы программы.

9. Описать класс с именем KNIGA, содержащий следующие поля:

- Фамилия автора;
- Название книги;
- Издательство;
- Год издания;
- Количество страниц.

а) Дополнительно введите поля название библиотеки и номер дома, где расположена здание библиотеки. С помощью конструктора при инициализации массива объектов класса KNIGA установите значения этих полей равными по умолчанию «Oblastnay im Nikitina» и 15 соответственно.

б) Создайте деструктор, который обнулит поле номер дома и очистит поле название библиотеки, для объектов, у которых количество страниц менее 200.

Выведите на экран результаты работы программы.

## Пример выполнения задания 1

```
#include "stdafx.h"
#include <iostream>
#include <cstring>

using namespace std;

class C_pacient
{
public:
    C_pacient()
    {
        cnom_bol=7;
        char * p_nazv="ODKB";
        strcpy(cnazv_bol,p_nazv);
    }

    C_pacient (int nomer_bol)
    {
        cnom_bol=nomer_bol;
    };
    ~C_pacient();
    char* GetFam (void)
    {
        return cFam_in;
    };
    char* Getnazv_bol (void)
    {
        return cnazv_bol;
    };
    void SetFam (int i)
    {
        char newline;
        cout << "Vvedite FIO "<<i<<"-ogo pacienta: ";
        cin.get(cFam_in, 29, '\n');
        cin.get(newline);
    };

    void SetVozrast(int vozr)
    {
        cvozzrast=vozr;
    };
    int GetVvozzrast(void)
    {
        return cvozzrast;
    };
    int Getnom_bol(void)
    {
        return cnom_bol;
    };

private:
    char cFam_in[30];
    int cvozzrast;
    int cnom_bol;
```

```

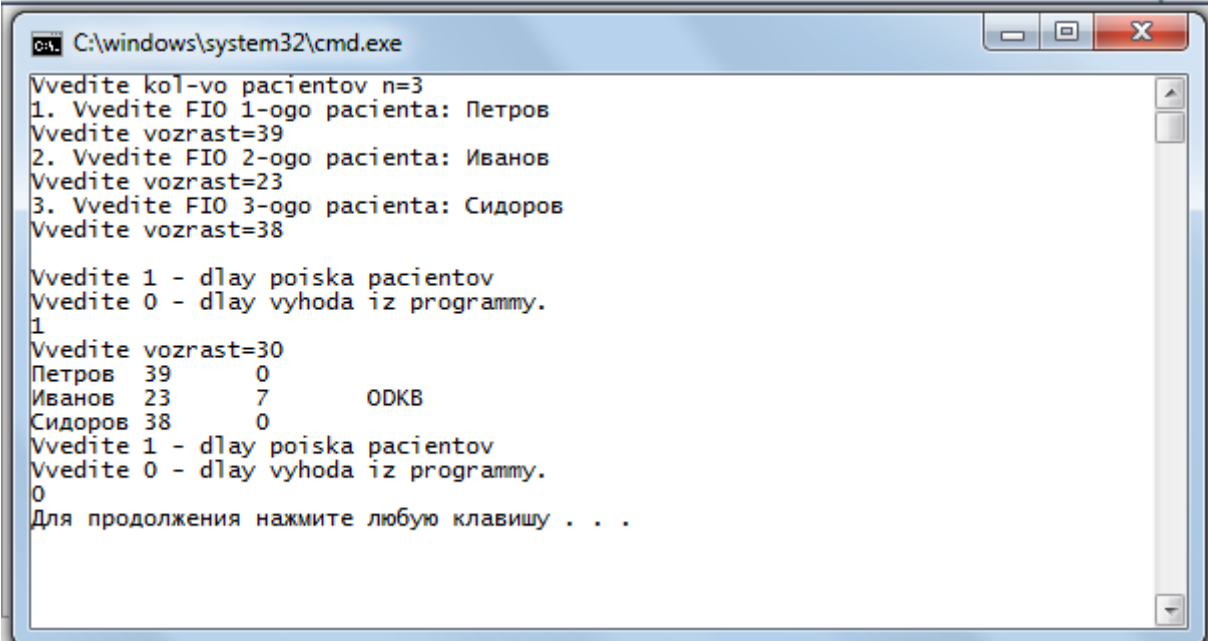
        char cnazv_bol[20];
    };

C_pacient::~C_pacient()
{
    cnom_bol=0;
    strcpy(cnazv_bol,"");
}

int main()
{ char newline;
  C_pacient bolnye[30];
  int n,m;
  int voz;
  cout<<"Vvedite kol-vo pacientov n=";
  cin>>n;
  cin.get(newline);
      for (int i=0; i<n; i++)
      {
          cout<<i+1<<" ";
          bolnye[i].SetFam(i+1);
          cout <<"Vvedite vozrast=";
          cin >>voz;
          cin.get(newline);
          bolnye[i].SetVozrast(voz);
      }
  cout << endl;
  int s=1;
  double vozr;
  while (s)
  {
      cout<<"Vvedite 1 - dlay poiska pacientov"<<endl;
      cout<<"Vvedite 0 - dlay vyhoda iz programmy."<<endl;
      cin>>s;
      cin.get(newline);
      if (s==0) return 0;
      cout<<"Vvedite vozrast=";
      cin>>vozr;
      for (int i=0; i<n; i++)
          if (bolnye[i].GetVvozrast()>vozr) bolnye[i].~C_pacient();

  for (int i=0; i<n; i++)
  {cout<<bolnye[i].GetFam()<<"\t"<<bolnye[i].GetVvozrast()<<"\t";
  cout<<bolnye[i].Getnom_bol()<<"\t"<<bolnye[i].Getnazv_bol()<<endl;
  }
  }
}

```



```
C:\windows\system32\cmd.exe
Vvedite kol-vo pacientov n=3
1. Vvedite FIO 1-ogo pacienta: Петров
Vvedite vozrast=39
2. Vvedite FIO 2-ogo pacienta: Иванов
Vvedite vozrast=23
3. Vvedite FIO 3-ogo pacienta: Сидоров
Vvedite vozrast=38

Vvedite 1 - dlay poiska pacientov
Vvedite 0 - dlay vyhoda iz programmy.
1
Vvedite vozrast=30
Петров 39      0
Иванов 23      7      ОДКВ
Сидоров 38     0
Vvedite 1 - dlay poiska pacientov
Vvedite 0 - dlay vyhoda iz programmy.
0
Для продолжения нажмите любую клавишу . . .
```

Рис. 1 Результаты работы программы

### Контрольные вопросы

1. Дать определение понятиям класс, объект.
2. Основные концепции объектно-ориентированного программирования.
3. Инкапсуляция, наследование, полиморфизм.
4. Объектно-ориентированный подход к разработке программ.
5. История и основы языка C++.
6. Примеры простых программ на языке C++.
7. Объектно-ориентированные средства языка C++.
8. Объекты, классы.
9. Инкапсуляция данных и методы доступа?
10. Конструктор класса.
11. Деструктор класса.