Федеральное агентство связи

Федеральное государственное бюджетное образовательное

учреждение высшего образования

Поволжский государственный университет

телекоммуникаций и информатики

Лабораторная работа №2

По дисциплине «Алгоритмы и структуры данных»

Тема: «Структурированные типы данных и массивы»

Выполнил студент

группы ИСТ-82 Логинов Ю.

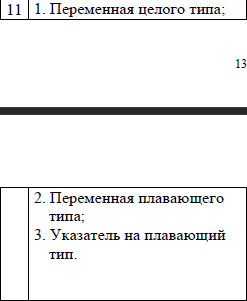
Проверил

Назаренко П. А.

Самара 2021

***Цель работы***

Получить навыки работы со структурированными типами, их объектами и компонентами, а также массивами объектов, в том числе динамическими. Научиться использовать объекты структурных типов в функциях, изучить способы передачи объектов в функции. Закрепить знания о динамических массивах.



1. Определить структурный тип, содержащий следующие поля:

– символьный массив, используемый для хранения строки, например, с именем студента,

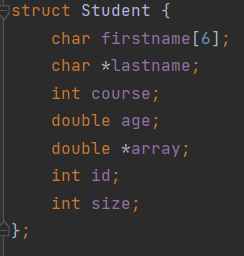
– указатель на тип **char** – для организации динамического массива, хранящего строку, например, с фамилией студента.

Остальные поля выбрать по вариантам, приведенным в таблице 2.

Использовать одну из переменных для хранения некоторого идентификатора (номера);

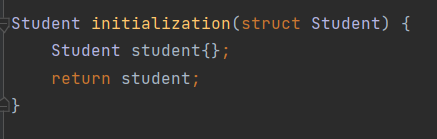
указатель на не символьный тип – для организации динамического массива целых или плавающих чисел; другую переменную – для хранения размера этого массива.

Дополнить структурный тип любыми полями по своему выбору.

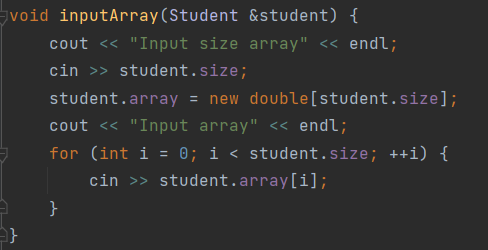


2. Определить функции:

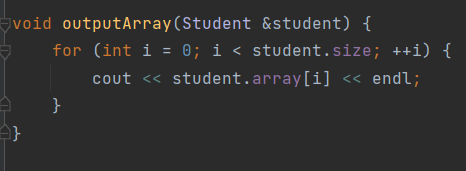
– инициализации структуры;



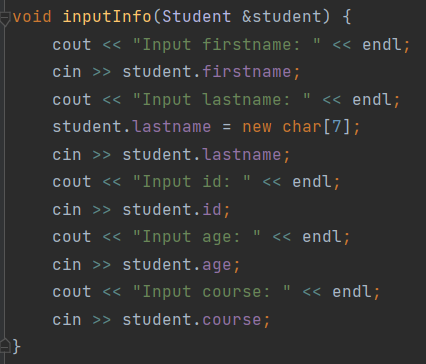
– заполнения массива чисел;



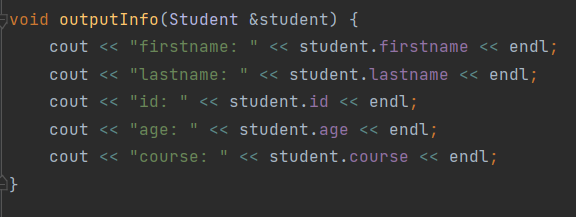
– вывода на экран массива чисел;



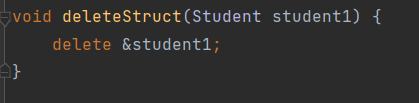
– ввода информации в строки имени и фамилии и другие поля;



– вывода на экран всех полей структуры, кроме массива чисел;

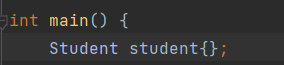


– функцию освобождения динамической памяти.



3. Определить функцию main(), в которой создать:

– объект ранее определенного структурного типа

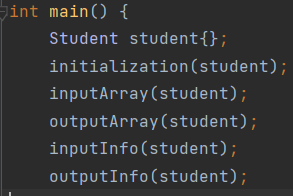


– указатель на этот структурный тип.

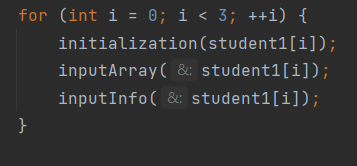


С помощью указателя создать динамический массив объектов структурного типа из 3-х – 4-х элементов.

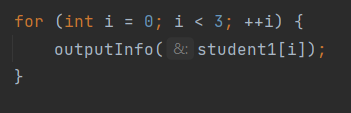
Для объекта последовательно вызывать функции инициализации, заполнения массива чисел, ввода данных в остальные поля, показа массива, показа полей.



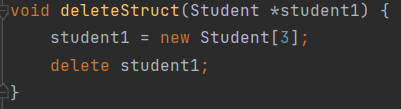
Для каждого элемента массива структур выполнить в цикле (for) функции инициализации, заполнения массива и ввода данных.



Вывести на экран содержимое полей каждого элемента массива структур в цикле (for) с помощью соответствующих функций.

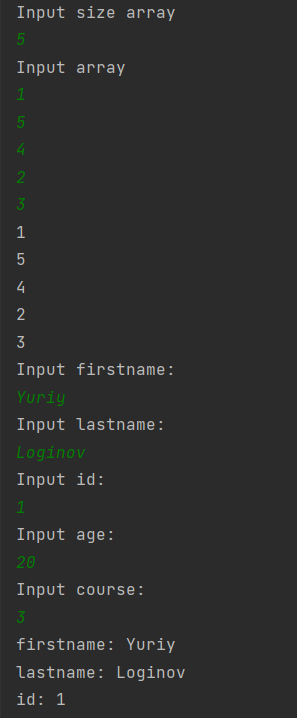


В конце функции main() вызвать функцию освобождения памяти для объекта структурного типа и в цикле для каждого элемента массива объектов.

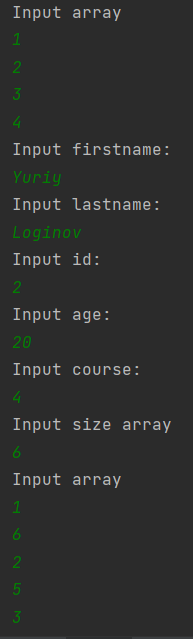


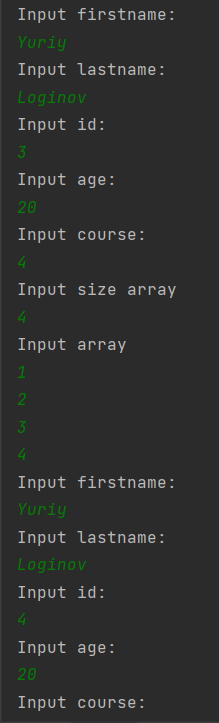
Ход выполнения программы контролировать по выводимым на экран сообщениям.

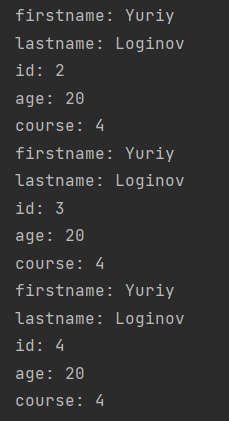
Ввод и вывод информации об объекте:



Ввод и вывод для каждого элемента массива структур:







Программа завершилась без ошибок:



Листинг программы:

#include <iostream>  
  
using namespace std;  
  
struct Student {  
 char firstname[6];  
 char \*lastname;  
 int course;  
 double age;  
 double \*array;  
 int id;  
 int size;  
};  
  
Student initialization(struct Student) {  
 Student student{};  
 return student;  
}  
  
void inputArray(Student &student) {  
 cout << "Input size array" << endl;  
 cin >> student.size;  
 student.array = new double[student.size];  
 cout << "Input array" << endl;  
 for (int i = 0; i < student.size; ++i) {  
 cin >> student.array[i];  
 }  
}  
  
void outputArray(Student &student) {  
 for (int i = 0; i < student.size; ++i) {  
 cout << student.array[i] << endl;  
 }  
}  
  
void inputInfo(Student &student) {  
 cout << "Input firstname: " << endl;  
 cin >> student.firstname;  
 cout << "Input lastname: " << endl;  
 student.lastname = new char[7];  
 cin >> student.lastname;  
 cout << "Input id: " << endl;  
 cin >> student.id;  
 cout << "Input age: " << endl;  
 cin >> student.age;  
 cout << "Input course: " << endl;  
 cin >> student.course;  
}  
  
void outputInfo(Student &student) {  
 cout << "firstname: " << student.firstname << endl;  
 cout << "lastname: " << student.lastname << endl;  
 cout << "id: " << student.id << endl;  
 cout << "age: " << student.age << endl;  
 cout << "course: " << student.course << endl;  
}  
  
void deleteStruct(Student \*student1) {  
 student1 = new Student[3];  
 delete student1;  
}  
  
int main() {  
 Student student{};  
 initialization(student);  
 inputArray(student);  
 outputArray(student);  
 inputInfo(student);  
 outputInfo(student);  
  
 Student \*student1;  
 student1 = new Student[3];  
  
 for (int i = 0; i < 3; ++i) {  
 initialization(student1[i]);  
 inputArray(student1[i]);  
 inputInfo(student1[i]);  
 }  
  
 for (int i = 0; i < 3; ++i) {  
 outputInfo(student1[i]);  
 }  
   
 for (int i = 0; i < 3; ++i) {  
 deleteStruct(&student1[i]);  
 }  
  
 return 0;  
}

**Вывод**: В данной работе мы получили навыки работы со структурированными типами, их объектами и компонентами, а также с массивами объектов. Научили использовать объекты структурного типа в функциях, изучили способы передачи объектов в функции.