Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине: «ЯП»

Тема: «КЛАССЫ И ОБЪЕКТЫ В С++»

Выполнил:

Студент 2 курса

Группы ПО-7(2)

Мозоль И.П.

Проверил:

Бойко Д.О.

2021

Цель работы: Получить практические навыки реализации классов на С++..

Вариант 1

Задание:

1. Определить пользовательский класс в соответствии с вариантом

задания (смотри приложение).

2. Определить в классе следующие конструкторы: без параметров, с

параметрами, копирования.

3. Определить в классе деструктор.

4. Определить в классе компоненты-функции для просмотра и уста-

новки полей данных.

5. Определить указатель на компоненту-функцию.

6. Определить указатель на экземпляр класса.

7. Написать демонстрационную программу, в которой создаются и

разрушаются объекты пользовательского класса и каждый вызов конст-

руктора и деструктора сопровождается выдачей соответствующего сооб-

щения (какой объект какой конструктор или деструктор вызвал).

8. Показать в программе использование указателя на объект и указа-

теля на компоненту-функцию.

Код программы:

“Student.h”

#ifndef STUDENT\_H  
#define STUDENT\_H  
#pragma once  
#include <iostream>  
#include <string>  
using namespace std;  
class Student{  
 string name;  
 int course;  
 bool gender;  
public:  
 Student();  
 Student(string stud\_name,int stud\_course,bool stud\_gender);  
 Student(const Student&stud);  
 ~Student();  
 string check\_gender();  
 string getName();  
 int getCourse();  
 bool getGender();  
 void show();  
 void setName(string stud\_name);  
 void setCourse(int stud\_course);  
 void setGender(bool stud\_gender);  
 void setData(string stud\_name,int stud\_course,bool stud\_gender);  
};  
#endif //LABA\_1\_YAP\_C\_\_\_STUDENT\_H

“Student.cpp”

#include "Student.h"  
Student::Student():  
 name("Ahmed"),course(1),gender(1)  
{  
 cout<<"Constructor with no parametres worked for \n"<<  
 name<<" "<<course<<" course " << check\_gender()<<'\n';  
}  
Student::Student(string stud\_name,int stud\_course,bool stud\_gender):  
name(stud\_name),course(stud\_course),gender(stud\_gender)  
{  
 cout<<"Constructor with parametres worked for \n"<<  
 name<<" "<<course<<" course " << check\_gender()<<'\n';  
}  
Student::Student(const Student&stud):  
 name(stud.name),course(stud.course),gender(stud.gender)  
{  
  
 cout<<"Constructor copy worked for\n"<<  
 name<<" "<<course<<" course " << check\_gender()<<'\n';  
}  
Student::~Student(){  
 cout<<"Destructor worked for\n"<<  
 name<<" "<<course<<" course " << check\_gender()<<'\n';  
}  
string Student::check\_gender(){  
 if(gender==1){  
 return "Man";  
 }  
 else return "Woman";  
}  
string Student::getName(){  
 return name;  
}  
int Student::getCourse(){  
 return course;  
}  
bool Student::getGender(){  
 return gender;  
}  
void Student::show(){  
 cout<<name<<" "<<course<<" course " << check\_gender()<<'\n';  
}  
void Student::setName(string stud\_name){  
 name=stud\_name;  
}  
void Student::setCourse(int stud\_course){  
 course=stud\_course;  
}  
void Student::setGender(bool stud\_gender){  
 gender=stud\_gender;  
}  
void Student::setData(string stud\_name,int stud\_course,bool stud\_gender){  
 name=stud\_name;  
 course=stud\_course;  
 gender=stud\_gender;  
}

“main.cpp”

#include "Student.h"  
int main(){  
 Student first;  
 Student second("Maria",4,0);  
 Student third(first);  
  
 void(Student::\*ShowShow)()=&Student::show;  
 (first.\*ShowShow)();  
  
 Student\* fourth=new Student;  
 cout <<fourth->getName()<<'\n';  
 fourth->setData("Tolik",5,1);  
 fourth->show();  
  
  
  
}

Вывод: В ходе выполнения данной работы я написал программу, в которой создаются и разрушаются объекты. Также я выполнил исследование вызовов конструкторов и деструкторов.