МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Белорусский государственный университет транспорта»

Кафедра «Информационно-управляющие системы и технологии»

Отчет  
по лабораторным работам

по дисциплине «Объектно-ориентированное программирование»

Выполнил Проверил

студент группы ГИ-21 ст. преп. каф. «ИУСиТ»

Зайцев И. С. Голдобина Т. А.

Гомель, 2020

Оглавление

[2 Лабораторная работа № 2 Тема: «Классы и объекты» 3](#_Toc51409034)

[1.1 Задания 3](#_Toc51409035)

[1.2 Контрольные вопросы 4](#_Toc51409036)

# Лабораторная работа № 2 Тема: «Классы и объекты»

### **Цель**

Изучить структуру класса, атрибуты доступа к компонентам класса; рассмотреть принцип работы конструкторов (с параметрами, с параметрами по умолчанию), конструктора копирования и деструктора при работе с объектом, статические и константные данные и методы.

## Задания

### Задание 1.1

### Условие

Создать класс «Окружность». Данные класса: радиус окружности.

Функции-члены класса: вычисление площади, вычисление длины окружности,

установление радиуса окружности и возвращение значения радиуса

окружности, вывод данных на экран. Функции-члены установки полей класса

должны проверять корректность задаваемых параметров.

Программный код

#include <iostream>

#include <math.h>

#define PI 3.141593

using namespace std;

class Circle{

public:

float r;

Circle(){

cout << "Enter the radius of the circle: ";

cin >> r;

};

int check()

{

if(r<=0)

{

cout << "Error!!!\n";

return -1;

}

}

float area()

{

float s;

s=PI\*pow(r,2);

cout << "Area = " << s <<endl;

return s;

}

float length()

{

float p;

p=2\*PI\*r;

cout << "Length = " << p <<endl;

return p;

}

};

int main()

{ int a;

///класс объект-имя создание-конструктор

Circle circle = Circle();

a=circle.check();

if(a==-1)

{

return -1;

}

else

{

circle.area();

circle.length();

}

return 0;

}

### Результат



Рисунок 1 – Результат задания 1

## Контрольные вопросы

1. В чем разница между struct, class и union?

В языке С++ класс, определяемый посредством ключевых слов struct, class, union, включает в себя функции и данные, создавая новый тип объектов. Компоненты класса имеют ограничения на доступ. Эти ограничения определяются ключевыми словами private, protected, public. Для ключевого слова class по умолчанию все компоненты будут private. Элементы класса типа структуры (struct) и объединения (union) по умолчанию принимаются как public.

1. Что такое указатель this? Приведите пример использования этого указателя.

Указатель this – скрытый указатель, который адресует начало выделенной под объект памяти. Для любой функции, принадлежащей классу my\_class, указатель this неявно объявлен так:

my\_class \*const this;

Пример:

class Something {

private:

    int data;

public:

    Something(int data) {

        this->data = data;  }};

1. Какова основная форма конструктора копирования и когда он вызывается?

Общий вид конструктора копирования имеет следующий вид:

имя\_класса (const имя\_класса & );

Когда объект копируется из функции main() в функцию, то должен вызываться конструктор для копирования. Т.е., если в конструкторе некоторого класса осуществляется динамическое выделение памяти, такой класс должен иметь соответствующий конструктор для копирования, а также деструктор (для освобождения памяти).

1. Когда вызывается деструктор?

Деструктор вызывается при уничтожении объекта.

1. Приведите пример использования константных и статических данных и методов.

#include <iostream>

using namespace std;

class cls

{

int kl; // количество изделий

double zp; // зарплата на производство одного изделия

double nl1,nl2; // два налога на зарплату

double sr; // количество сырья на производство одного изделия

static double cs; // цена сырья на одно изделие

public:

cls(){} // конструктор по умолчанию

~cls(){} // деструктор

void inpt(int);

static void vvod\_cn(double);

double seb() const;

};

double cls::cs; // явное определение static-члена в контексте файла

void cls::inpt(int k)

{

kl=k;

cin >> nl1 >> nl2 >> zp;

}

void cls::vvod\_cn(double c)

{

cs=c; // можно обращаться в функции только к static-компонентам

}

double cls::seb() const

{

return kl\*(zp+zp\*nl1+zp\*nl2+sr\*cs);

// в функции нельзя изменить ни один

// неявный параметр (kl zp nl1 nl2 sr)

}

int main()

{

setlocale(LC\_ALL,"Russian");

cls c1,c2;

cout << "Введите зарплату и два налога для объекта с1" << endl;

c1.inpt(100); // инициализация первого объекта

cout << "Введите зарплату и два налога для объекта с2" << endl;

c2.inpt(200); // инициализация второго объекта

cls::vvod\_cn(500.);

cout << "\nc1 = " << c1.seb() << "\nc2 = " << c2.seb() << endl;

return 0;

}

### Вывод по практической работе

Изучил структуру класса, атрибуты доступа к компонентам класса; рассмотрел принцип работы конструкторов (с параметрами, с параметрами по умолчанию), конструктора копирования и деструктора при работе с объектом, статические и константные данные и методы.