Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**О Т Ч Е Т**

**по лабораторной работе “Классы” №1**

**по дисциплине**

**«Основы алгоритмизации и программирования» семестр 2**

Выполнил студент гр. ИВТ-21-1б

Ипатов Дмитрий Сергеевич

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Д.В.

(оценка) (подпись)

г. Пермь-2022

**Постановка задачи:**

1. Реализовать определение нового класса. Для демонстрации работы с объектами написать главную функцию. Продемонстрировать разные способы создания объектов и массивов объектов.
2. Структура-пара – структура с двумя полями, которые обычно имеют имена first и second. Требуется реализовать тип данных с помощью такой структуры. Во всех заданиях должны присутствовать:
   1. метод инициализации Init (метод должен контролировать значения аргументов на корректность);
   2. ввод с клавиатуры Read;
   3. вывод на экран Show.

1. Реализовать внешнюю функцию make\_тип, где тип – тип реализуемой структуры. Функция должна получать значения для полей структуры как параметры функции и возвращать структуру как результат. При передаче ошибочных параметров следует выводить сообщение и заканчивать работу.

**Анализ задачи:**

1. Класс является абстрактным типом данных, определяемым пользователем, и представляет собой модель реального объекта в виде данных и функций для работы с ними.
2. Данные класса называются полями (по аналогии с полями структуры) или атрибутами, а функции класса — методами. Поля и методы называются элементами класса.
3. Спецификаторы доступа private и public управляют видимостью элементов класса. Элементы, описанные после служебного слова private, видимы только внутри класса. Этот вид доступа принят в классе по умолчанию. Интерфейс класса описывается после спецификатора public. Действие любого спецификатора распространяется до следующего спецификатора или до конца класса. Можно задавать несколько секций private и public, порядок их следования значения не имеет.
4. Получить информацию о содержимом полей, описанных после спецификатора private можно только с помощью специальных методов, которые называются селекторами, а изменить – с помощью методов, которые называются модификаторами.
5. Видимостью элементов класса можно также управлять с помощью ключевых слов struct и class. Если при описании класса используется слово struct, то все поля и методы по умолчанию будут общедоступными (public). Если при описании класса используется слово class, то по умолчанию все методы и поля класса будут скрытыми (private).

**Код программы:**

#include <iostream>

#include <sstream>

#include <cmath>

using namespace std;

class nums

{

private:

double first;

double second;

double k;

public:

void read()

{

cout << "Введите целое число: " << endl;

cin >> first;

cout << "Введите дробное число:" << endl;

cin >> second;

cout << "Введите вещественное число (Пример: 324.32):" << endl;

cin >> k;

init();

}

void init()

{

if ((first != round(first) or first < 0) or (second == round(second) or second < 0) or (k==round(k)))

{

cout << "ОШИБКА!!! Числа введены неправильно!";

abort();

}

else

{

cout << "Числа введены правильно. Выполняем умножение" << endl;

multiply(k);

}

}

void show()

{

cout << "Результат умножения целого на вещественное: " << first << " Результат умножения дробного на вещественное: " << second << endl;

}

void multiply(double k)

{

first \*= k;

second \*= k;

}

};

nums makeNums(const nums& b)

{

double first, second, k;

cout << "Введите целое число: " << endl;

cin >> first;

cout << "Введите дробное число:" << endl;

cin >> second;

cout << "Введите вещественное число (Пример: 324.32):" << endl;

cin >> k;

if ((first != round(first) or first < 0) or (second == round(second) or second < 0) or (k == round(k)))

{

cout << "ОШИБКА!!! Числа введены неправильно!";

abort();

}

else

{

cout << "Числа введены правильно. Выполняем умножение" << endl;

first \*= k;

second \*= k;

cout << "Результат умножения целого на вещественное: " << first << " Результат умножения дробного на вещественное: " << second;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

nums a;

a.read();

a.show();

nums b;

makeNums(b);

}

**Скриншоты результатов:**

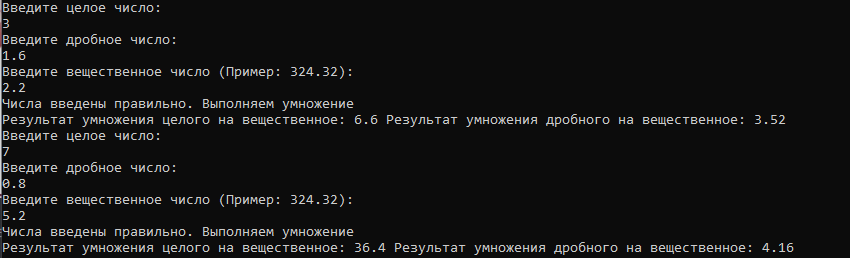
****

Рис. 1 Результаты программы при правильном вводе переменных

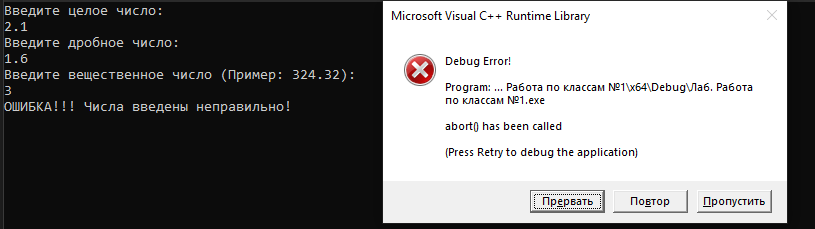
****

Рис. 2 Результаты программы при неправильном вводе переменных. Выводится ошибка

**Контрольные вопросы:**

1. Что такое класс?  
Класс – абстрактный тип данных, определяемый пользователем. Представляет собой модель реального объекта в виде данных и функций.  
2. Что такое объект (экземпляр) класса?  
Объект – переменная класса.  
3. Как называются поля класса?  
Полями класса называют данные, содержащиеся в классе.  
4. Как называются функции класса?  
Функциями класса называются методы.  
5. Для чего используются спецификаторы доступа?  
Спецификаторы доступа используются для изменения видимости некоторых компонентов класса.  
6. Для чего используется спецификатор public?  
Public используется для того, чтобы компоненты класса были открыты к доступу извне.  
7. Для чего используется спецификатор private?  
Private используется для того, чтобы компоненты класса были закрыты от доступа извне.  
8. Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?  
Private.  
9. Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использоваться по умолчанию?  
Public.  
10. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?  
Должен использоваться спецификатор public для того, чтобы к методу можно было обратиться извне.  
11. Каким образом можно изменить значения атрибутов экземпляра класса?  
Изменить значение можно через методы, описанные в public, или, если поля публичные, обратиться к ним напрямую.  
12. Каким образом можно получить значения атрибутов экземпляра класса?  
Получить значение можно через методы, описанные в public, или, если поля публичные, обратиться к ним напрямую.  
13. Класс описан следующим образом:  
Struct Student  
{  
String name;  
Int group;  
…  
};  
Объект класса определен следующим образом  
Student \*s = new Student;  
Как можно обратиться к полю name объекта s?  
  
S[i].name = “”;// i – номер элемента  
  
14.  
Struct Student  
{  
String name;  
Int group;  
…  
};  
Student s;  
Как можно обратиться к полю name объекта s?  
[s.name](https://vk.com/away.php?to=http%3A%2F%2Fs.name&cc_key=) = “”;  
  
15. class Student  
{  
String name;  
Int group;  
…  
};  
Student \*s = new Student;  
Можно обратиться только через методы класса.  
  
16.  
class Student  
{  
String name;  
Int group;  
…  
};  
Объект класса определен следующим образом  
Student s;  
Как можно обратиться к полю name объекта s?  
  
Можно обратиться только через методы класса.  
17. Класс описан следующим образом:  
class Student  
{  
Public:  
String name;  
Int group;  
…  
};  
Объект класса определен следующим образом:  
Student\* s = new Student;