Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №18.1**

Дисциплина: «Информатика»

Тема: Классы и объекты. Инкапсуляция

Вариант 12

Выполнил:

Студент группы РИС-20-2б

Пономарев Егор Витальевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

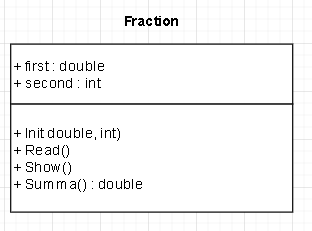
1. Создание консольного приложения из нескольких файлов в системе программирования Visual Studio.
2. Использование классов и объектов в ОО программе.

**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:
3. Метод инициализации Init (метод должен контролировать значения аргументов на корректность);
4. Ввод с клавиатуры Read;
5. Вывод на экран Show.
6. 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.

**Задание варианта:** поле first – дробное положительное число, оклад, поле second – целое положительное число, количество отработанных дней. Реализовать метод summa () – вычисление начисленной суммы за данное количество дней по формуле: оклад/количество\_дней\_месяца\*количество\_отработанных\_дней

**UML диаграмма**



**Код C++**

**Fraction.h:**

#pragma once

#icnlude <iostream>

struct fraction {

double first;

int second;

void Init(double, int); // метод для инициализации полей

void Read(); // метод для чтения значений полей

void Show(); // метод для вывода значений полей

double Summa(); // вычисление степени

};

**Fraction.cpp:**

#include "fraction.h"

using namespace std;

void fraction::Init(double F, int S) {

while (F < 0 || S < 0) {

cout << "Ошибка. Введено отрицательное число. Попробуйте ещё раз: ";

cin >> F;

cin >> S;

}

first = F;

second = S;

}

void fraction::Read() {

cout << "\nfirst? ";

cin >> first;

while (first < 0) {

cout << "Ошибка. Введено отрицательное число. Попробуйте ещё раз: ";

cin >> first;

}

cout << "\nsecond? ";

cin >> second;

while (second < 0) {

cout << "Ошибка. Введено отрицательное число. Попробуйте ещё раз: ";

cin >> first;

}

}

void fraction::Show() {

cout << "\nfirst = " << first;

cout << "\nsecond = " << second;

cout << endl;

}

double fraction::Summa() {

return first \* second;

}

**Main.cpp**

#include "fraction.h"

using namespace std;

fraction make\_fraction(double F, int S) {

fraction t;

t.Init(F, S);

return t;

}

int main()

{

fraction A;

fraction B;

A.Init(3.0, 2);

B.Read();

A.Show();

B.Show();

cout << "A.Summa(" << A.first << ", " << A.second << ") = " << A.Summa() << endl;

cout << "B.Summa(" << B.first << ", " << B.second << ") = " << B.Summa() << endl;

fraction\* X = new fraction;

X->Init(2.0, 5);

X->Show();

X->Summa();

cout << "X.Summa(" << X->first << ", " << X->second << ") = " << X->Summa() << endl;

fraction mas[3];

for (int i = 0; i < 3; i++)

mas[i].Read();

for (int i = 0; i < 3; i++)

mas[i].Show();

for (int i = 0; i < 3; i++) {

mas[i].Summa();

cout << "mas[" << i << "].Summa(" << mas[i].first << ", " << mas[i].second << ") = ";

cout << mas[i].Summa() << endl;

}

fraction\* p\_mas = new fraction[3];

for (int i = 0; i < 3; i++)

p\_mas[i].Read();

for (int i = 0; i < 3; i++)

p\_mas[i].Show();

for (int i = 0; i < 3; i++) {

p\_mas[i].Summa();

cout << "p\_mas[" << i << "].Summa(" << p\_mas[i].first << ", " << p\_mas[i].second << ") = ";

cout << p\_mas[i].Summa() << endl;

}

double y;

int z;

cout << "first? ";

cin >> y;

cout << "second? ";

cin >> z;

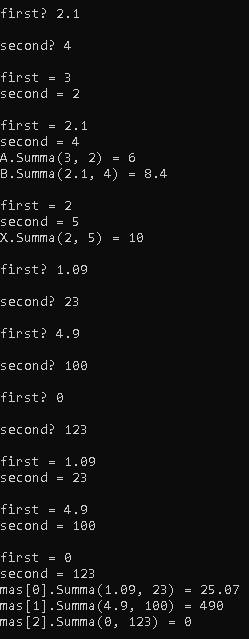
fraction F = make\_fraction(y, z);

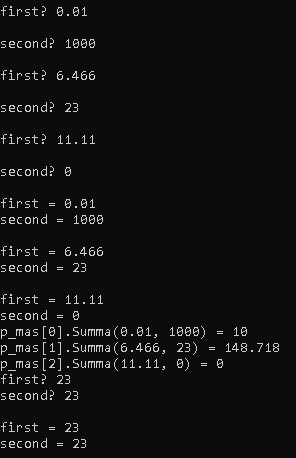
F.Show();

return 0;

}

**Результаты выполнения**





**Контрольные вопросы**

1. ***Что такое класс?***

Класс является абстрактным типом данных, определяемым пользователем, и представляет собой модель реального объекта в виде данных и функций для работы с ними.

1. ***Что такое объект (экземпляр) класса?***

Это отдельный представитель класса, имеющий конкретное состояние и поведение, полностью определяемое классом.

1. ***Как называются поля класса?*** – Атрибуты.
2. ***Как называются функции класса?*** – Методы.
3. ***Для чего используются спецификаторы доступа?***

Спецификаторы доступа управляют видимостью элементов класса.

1. ***Для чего используется спецификатор public?***

Спецификатор public используется для описания интерфейса класса.

1. ***Для чего используется спецификатор private?***

Элементы, описанные после служебного слова private, видимы только внутри класса. В основном используется для описания атрибутов.

1. ***Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?*** – Private.
2. ***Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использоваться по умолчанию?*** – Public.
3. ***Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?***

Для описания интерфейса должен использоваться спецификатор public, так как к этим элементам можно обращаться как через объект класса, так и через объект типа соответствующего интерфейса, в отличие от private.

1. ***Каким образом можно изменить значения атрибутов экземпляра класса?***

В конструкторе, напрямую (если public) или с помощью public метода.

1. ***Каким образом можно получить значения атрибутов экземпляра класса?***

С помощью public функции, которая возвращает значение атрибута.

1. 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?

1. Класс описан следующим образом

Struct Student

{

string name;

int group;

……..

};

Объект класса определен следующим образом

Student s;

Как можно обратиться к полю name объекта s?

1. Класс описан следующим образом

class Student

{

string name;

int group;

……..

};

Объект класса определен следующим образом

Student \*s = new Student;

Как можно обратиться к полю name объекта s?

1. Класс описан следующим образом

class Student

{

string name;

int group;

……..

};

Объект класса определен следующим образом

Student s;

Как можно обратиться к полю name объекта s?

1. Класс описан следующим образом

class Student

{

Public:

Char\* name;

int group;

……..

};

Объект класса определен следующим образом

Student \*s = new Student;

Как можно обратиться к полю name объекта s?

Поставленные задачи можно решать следующим образом:

|  |
| --- |
| Triangle Make\_Triangle(float first, float second) {  Triangle temp;  temp.Init(first, second);  return temp; } |

2.

|  |
| --- |
| void Triangle::ShowSides() {  cout << "First = " << first << " Second = " << second << endl; }  void Triangle::ShowHipotenuse() {  cout << "Hipotenuse = " << Hipotenuse() << endl; } |

3.

|  |
| --- |
| bool Triangle::Init(float first, float second) {  if (first <= 0 || second <= 0) return false;  else  {  this->first = first;  this->second = second;  }  } |

4.

|  |
| --- |
| void Triangle::Read() {  cout << "Enter first: "; cin >> first;  cout << "Enter second: "; cin >> second;  while (first <= 0 || second <= 0)  {  cout << "U've entered incorrect data:\n\n";  cout << "Enter first again: "; cin >> first;  cout << "Enter second again: "; cin >> second;  }  cout << endl; } |

5.

|  |
| --- |
| for (int i = 0; i < size; ++i)  {  if (obj\_arr[i].Init(0.153 \* pow((i + 1), i + 2), 0.151 \* pow((i + 2), i + 1)))  {  obj\_arr[i].ShowSides();  obj\_arr[i].Hipotenuse();  obj\_arr[i].ShowHipotenuse();  cout << endl;  }  } |