Министерство высшего образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

по лабораторной работе №1

Тема: «Классы и объекты. Инкапсуляция.»

Выполнил

Студент группы РИС-22-2б

Худеньких В.Д.

Проверил доц. Кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2023

# 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.
3. 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.

ВАРИАНТ 15:

Элемент арифметической прогрессии aj вычисляется по формуле: aj=a0·rj.Поле first – дробное число, первый элемент прогрессии a0, поле second – положительное целое число, постоянное отношение r. Реализовать метод element(int j) –вычисление j-го элемента прогрессии.

# Контрольные вопросы

* 1. Что такое класс?

*Класс - абстрактный тип данных, определяемым пользователем. Представляет собой модель реального объекта в виде данных и функций для работы с ними.*

* 1. Что такое объект (экземпляр) класса?

*Экземпляр класса – это описание конкретного объекта класса. Допустим: Класс – рыба, объект – окунь.*

* 1. Как называются поля класса?

*Атрибуты.*

* 1. Как называются функции класса?

*Методы.*

* 1. Для чего используются спецификаторы доступа?

*Управление видимостью элементов класса (чтение/запись).*

* 1. Для чего используется спецификатор public?

*Для общедоступных элементов класса*

* 1. Для чего используется спецификатор private?

*Для закрытых элементов класса.*

* 1. Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?

*private*

* 1. Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использоваться по умолчанию?

*public*

* 1. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?

*Public. Так как нужно иметь доступ к методу, чтобы его описать в дальнейшем.*

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

*С помощью метода – модификатора (сеттер).*

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

*С помощью метода – селектор (геттер).*

* 1. *s->name;*
  2. *s.name;*
  3. *с помощью геттера*
  4. *с помощью геттера*
  5. *s->name.*

***Код программы***

*Заголовочный файл, который содержит описание класса (fraction.h)*

#include <iostream>

using namespace std;

struct **fraction**

{

double first;

int second;

void **Init**(double, int);

void **Read**();

void **Show**();

double **Element**(int j);

};

*Файл, содержащий описание методов класса fraction (fraction.cpp)*

#include "fraction.h"

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

{

first = F;

second = S;

}

void fraction::**Read**()

{

cout << "\nfirst?"; cin >> first;

cout << "\nsecond?"; cin >> second;

}

void fraction::**Show**()

{

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

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

cout << "\n";

}

double fraction::**Element**(int j)

{

return first \* pow(second, j);

}

*Файл, содержащий основную программу (main.cpp)*

#include "fraction.h"

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.Power(" << A.first << "," << A.second << ")=" << A.Element(5) << endl;

cout << "B.Power(" << B.first << "," << B.second << ")=" << B.Element(5) << endl;

fraction\* X = new fraction;

X->Init(2.0, 5);

X->Show();

X->Element(5);

cout << "X.Element(" << X->first << "," << X->second << ")=" << X->Element(5) << endl;

delete X;

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].Element(5);

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

cout << mas[i].Element(5) << 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].Element(5);

cout << "p\_mas[" << i << "].Power(" << p\_mas[i].first << "," << p\_mas[i].second;

cout << ")=" << p\_mas[i].Element(5) << endl;

}

delete[] p\_mas;

double y; int z;

cout << "first?"; cin >> y;

cout << "second?"; cin >> z;

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

F.Show();

system("pause");

return 0;

}