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, объекты и их представление в ОО программировании

Вариант 1

Выполнил:

Студент группы РИС-20-1б

Азмагулов Артём Вадимович

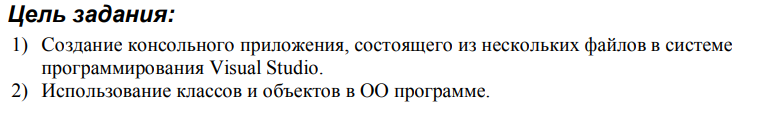
Проверила:

Доцент кафедры ИТАС

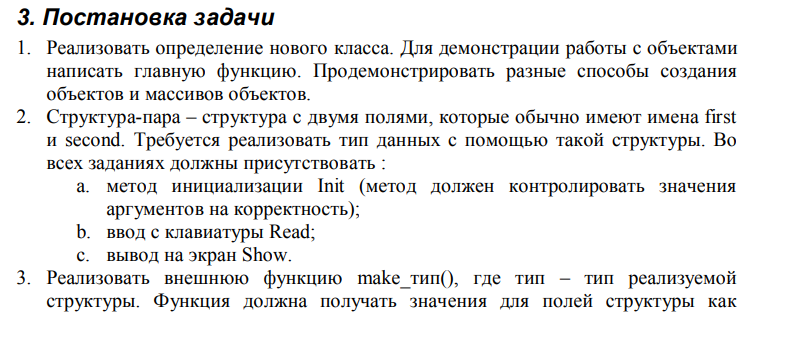
Полякова О. А.

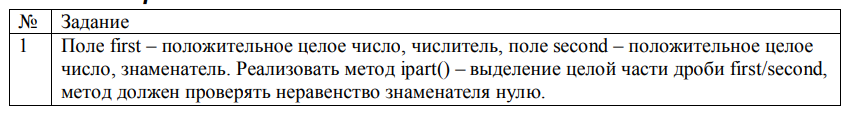
**Пермь, 2021**

**Цель работы**

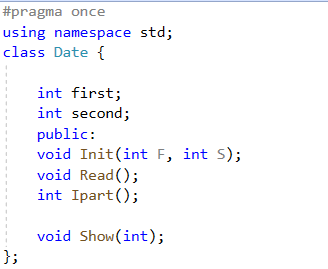


**Постановка задачи**

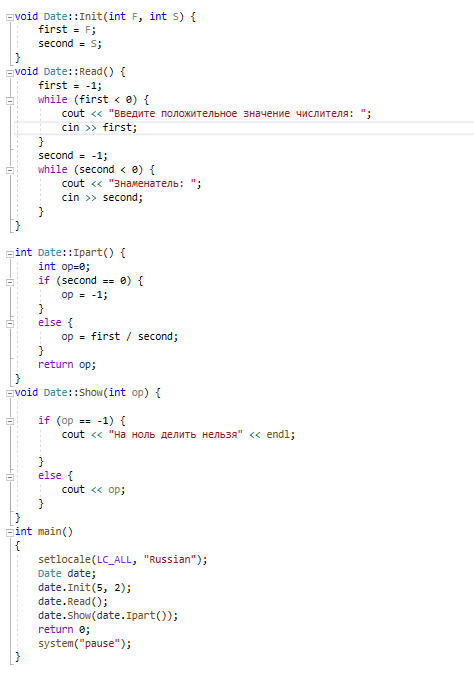


**Анализ задачи**

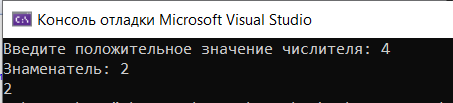
1. Для решения данной задачи необходимо:
   1. Создать класс Date с атрибутами и методами:



* 1. Реализовать методы класса:



**Результаты работы программы**



**Решение**

Date.h:

#pragma once

using namespace std;

class Date {

int first;

int second;

public:

void Init(int F, int S);

void Read();

int Ipart();

void Show(int);

};

18.1.cpp:

#include <iostream>

#include "Data.h"

using namespace std;

void Date::Init(int F, int S) {

first = F;

second = S;

}

void Date::Read() {

first = -1;

while (first < 0) {

cout << "Введите положительное значение числителя: ";

cin >> first;

}

second = -1;

while (second < 0) {

cout << "Знаменатель: ";

cin >> second;

}

}

int Date::Ipart() {

int op=0;

if (second == 0) {

op = -1;

}

else {

op = first / second;

}

return op;

}

void Date::Show(int op) {

if (op == -1) {

cout << "На ноль делить нельзя" << endl;

}

else {

cout << op;

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

Date date;

date.Init(5, 2);

date.Read();

date.Show(date.Ipart());

return 0;

system("pause");

}

**Ответы на вопросы:**

1. Контрольные вопросы
2. Что такое класс?

- абстрактный тип данных, определяемый пользователем, и представляет собой модель реального объекта в виде данных и функций для работы с ними

1. Что такое объект класса?

- переменная класса

1. Как называются поля класса?

- атрибутами

1. Как называются функции класса?

- методами

1. Для чего используются спецификаторы доступа?

- для управления видимостью элементов класса

1. Для чего используется спецификатор public?

- для того, чтобы сделать поля и методы класса доступными вне класса

1. Для чего используется спецификатор private?

- для того, чтобы сделать элементы класса видимыми только внутри класса

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

- по умолчанию поля и методы будут скрытыми

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

- по умолчанию поля будут общедоступными

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

- должен использоваться спецификатор public, т.к. интерфейс может использоваться в других классах

1. Каким образом можно изменить значения атрибутов экземпляра класса?

- обратиться через точку к нужному атрибуту экземпляра и присвоить нужное значение атрибуту (если он в классе общедоступный)

1. Каким образом можно получить значения атрибутов экземпляра класса?

- обратиться через точку к нужному атрибуту экземпляра (если он в классе общедоступный)

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?

* s->name;

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?

- к полю name не получиться обратиться, т.к. оно не является общедоступным

1. Класс описан следующим образом

class Student {

string name;

int group;

pulic:

…

};

Объект класса определен следующим образом

Student s;

Как можно обратиться к полю name объекта s?

- к полю name не получиться обратиться, т.к. оно не является общедоступным

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?

- s->name;

**UML-диаграмма**

