Министерство науки и высшего образования РФ

Федеральное бюджетное государственное образовательное учреждение высшего профессионального образования

Пермский национальный исследовательский политехнический университет (ПНИПУ)

Факультет: Электротехнический (ЭТФ)

Направление: 09.03.01 «Информатика и вычислительная техника (Автоматизированные системы обработки информации и управления)»

Кафедра: «Информационные технологии и автоматизированных систем» (ИТАС)

Информатика

Лабораторная работа №1

Классы и объекты. Инкапсуляция

Студент: Балтаев Э. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись)

Группа: АСУ-20-1бзу

Работу проверил:

доцент кафедры ИТАС

Полякова О. А.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Подпись:\_\_\_\_\_\_\_\_\_\_\_

Пермь 2021г.

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 структуры. Во всех заданиях должны присутствовать:

а. метод инициализации Init (метод должен контролировать значения аргументов на корректность);

b. ввод с клавиатуры Read;

c. 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.

Задание варианта:

Линейное уравнение y = Ax + B. Поле first – дробное число, коэффициент А, поле second – дробное число, коэффициент В. Реализовать метод function (double x) – вычисление значения y для заданного х.

Описание класса

#include <iostream>

#include <string>

using namespace std;

struct linear

{

double first;

double second;

void Init(double f, double s);

void Read();

void Show();

double Solve(double x);

};

Определение компонентных функций

// ошибок в значениях быть не может, коэффициенты любые числа

void linear::Init(double f, double s) {

first = f;

second = s;

}

void linear::Read() {

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

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

}

void linear::Show() {

cout << "y = " << first << "x ";

if (second > 0)

cout << " + " << second << endl;

else

cout << " - " << second << endl;

}

double linear::Solve(double x) {

return first \* x + second;

}

Определение главного файла

#include <iostream>

#include "linear.h"

linear make\_linear(double a, double b) {

linear l;

l.Init(a, b);

return l;

}

int main()

{

system("chcp 1251");

srand(time(0));

cout << "\nОбъект\n";

double res, x;

linear a;

a.Read();

a.Show();

cout << "x? "; cin >> x;

res = a.Solve(x);

cout << "res = " << res << endl;

cout << "\nМассив объектов\n";

const int N = 3;

linear arr[N];

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

arr[i].Init(rand() % 10 - 5, rand() % 10 - 5);

arr[i].Show();

}

cout << "\nИспользование функции\n";

double A, B;

cout << "A? "; cin >> A;

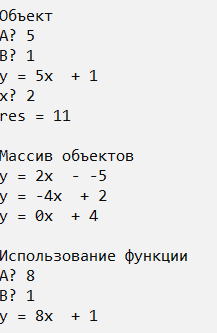
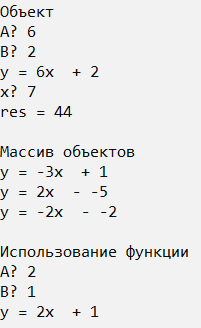
cout << "B? "; cin >> B;

linear b = make\_linear(A, B);

b.Show();

}

Результаты программы:

Ответы на контрольные вопросы:

1) Пользовательский тип данных

2) Переменная данного типа

3) Свойства объекта

4) Методы объекта

5) Чтобы скрыть некоторые поля и методы

6) Доступен везде

7) Доступен только внутри класса

8) private

9) public

10) public, чтобы мы могли использовать в других файлах

11) селекторы set<name>

12) геттеры get<name>

13) Не можем обратиться, он private

14) s.name

15) Не можем обратиться, он private

16) Не можем обратиться, он private

17) [ s🡪name ], [ (\*s).name ]