Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.2**

Дисциплина: основы алгоритмизации и программирования

Тема: “Классы и объекты. Использование конструкторов.”

Вариант 15

Выполнил:

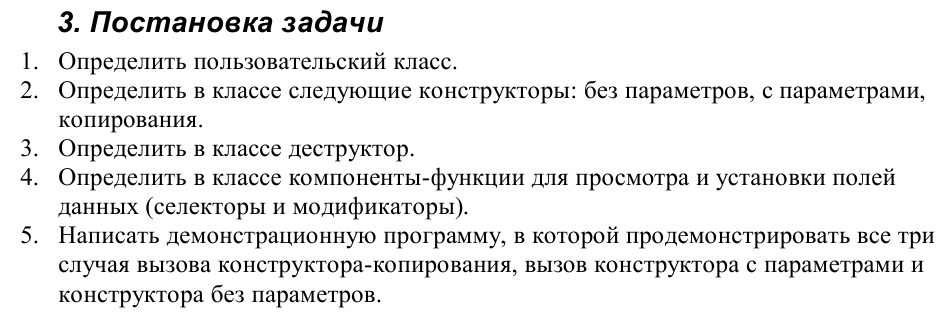
студент группы ИВТ-20-2Б Чувашев Максим

Проверила: доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Постановка задачи**





**Анализ задачи**

1. Используемые типы данных
   1. Double
   2. Int
   3. String
2. Действия над используемыми данными
   1. Int, double – арифметические операции.
3. Для решения задачи данные были представлены в следующем виде:
   1. Данные вводятся через объекты класса Tovar через геттеры.

class Salary

{

//атрибуты

private:

string FIO;

int Oklad;

double Prize;

public:

Salary();//конструктор без параметров

Salary(string, int, double);//конструктор с параметрами

Salary(const Salary&);//конструктор копирования

~Salary();//деструктор

string get\_FIO();//селектор

void set\_FIO(string);//модификатор

int get\_Oklad();//селектор

void set\_Oklad(int); //модификатор

double get\_Prize();//селектор

void set\_Prize(double); //модификатор

void show();//просмотр атрибутов

};

* 1. Данные так же вводятся через конструктор класса Tovar

//конструктор без параметров

Salary::Salary()

{

setlocale(LC\_ALL, "Ru");

FIO = "";

Oklad = 0;

Prize = 0;

cout << "Конструктор без параметров для объекта:\t" << this << endl;

}

//конструктор с параметрами

Salary::Salary(string N, int K, double S)

{

setlocale(LC\_ALL, "Ru");

FIO = N;

Oklad = K;

Prize = S;

cout << "Конструктор с параметрами для объекта:\t" << this << endl;

}

//конструктор копирования

Salary::Salary(const Salary &other)

{

setlocale(LC\_ALL, "Ru");

FIO = other.FIO;

Oklad = other.Oklad;

Prize = other.Prize;

cout << "Конструктор копирования для объекта:\t" << this << endl;

}

1. Для операций ввода и вывода использовались следующие операторы и функции:
   1. Для ввода числа используется сеттер

void Salary::set\_FIO(string N)

{

FIO = N;

}

void Salary::set\_Oklad(int K)

{

Oklad = K;

}

void Salary::set\_Prize(double S)

{

Prize = S;

}

* 1. Для получения числа используется геттер

string Salary::get\_FIO()

{

return FIO;

}

int Salary::get\_Oklad()

{

return Oklad;

}

double Salary::get\_Prize()

{

return Prize;

}

1. Поставленные задачи будут решены следующими действиями:
   1. В функции мейн происходит инициализации объектов класса с помощью конструктора с параметрами, с помощью конструктора без параметров и с помощью конструктора копирования

void main()

{

system("chcp 1251>nul");

setlocale(LC\_ALL, "Ru");

//конструктор без параметров

Salary t1;

t1.show();

//коструктор с параметрами

Salary t2("Чувашев Максим Алексеевич", 1,15000);

t2.show();

//конструктор копирования

Salary t3(t2);

t3.set\_FIO("Исламов Ильназ Максимович");

t3.set\_Oklad(53800);

t3.set\_Prize(10500.0);

//печать через конструктор копирования

print\_tovar(t3);

//конструктор копирования

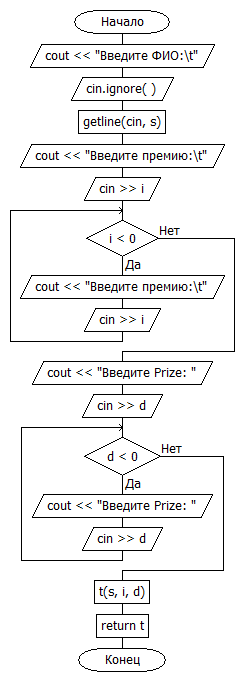
t1=make\_tovar();

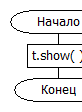
t1.show();

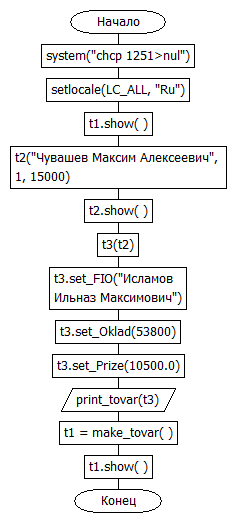
}

**Блок-схема**

**Sourse.cpp**







**Tovar.cpp**





**Tovar.h**



**Код**

#include "Tovar.h"

#include <iostream>

#include <string>

using namespace std;

//функция для возврата объекта как результата

Salary make\_tovar()

{

string s;

int i;

double d;

cout << "Введите ФИО:\t";

cin.ignore();

getline(cin, s);

cout << "Введите премию:\t";

cin >> i;

while (i < 0)

{

cout << "Введите премию:\t";

cin >> i;

}

cout << "Введите Prize: ";

cin >> d;

while (d < 0)

{

cout << "Введите Prize: ";

cin >> d;

}

Salary t(s, i, d); return t;

}

//функция для передачи объекта как параметра

void print\_tovar(Salary t)

{

t.show();

}

void main()

{

system("chcp 1251>nul");

setlocale(LC\_ALL, "Ru");

//конструктор без параметров

Salary t1;

t1.show();

//коструктор с параметрами

Salary t2("Чувашев Максим Алексеевич", 1,15000);

t2.show();

//конструктор копирования

Salary t3(t2);

t3.set\_FIO("Исламов Ильназ Максимович");

t3.set\_Oklad(53800);

t3.set\_Prize(10500.0);

//печать через конструктор копирования

print\_tovar(t3);

//конструктор копирования

t1=make\_tovar();

t1.show();

}

#include "Tovar.h"

#include <iostream>

#include <string>

using namespace std;

//конструктор без параметров

Salary::Salary()

{

setlocale(LC\_ALL, "Ru");

FIO = "";

Oklad = 0;

Prize = 0;

cout << "Конструктор без параметров для объекта:\t" << this << endl;

}

//конструктор с параметрами

Salary::Salary(string N, int K, double S)

{

setlocale(LC\_ALL, "Ru");

FIO = N;

Oklad = K;

Prize = S;

cout << "Конструктор с параметрами для объекта:\t" << this << endl;

}

//конструктор копирования

Salary::Salary(const Salary &other)

{

setlocale(LC\_ALL, "Ru");

FIO = other.FIO;

Oklad = other.Oklad;

Prize = other.Prize;

cout << "Конструктор копирования для объекта:\t" << this << endl;

}

//деструктор

Salary::~Salary()

{

setlocale(LC\_ALL, "Ru");

cout << "Деструктор для объекта:\t" << this << endl;

}

//селекторы

string Salary::get\_FIO()

{

return FIO;

}

int Salary::get\_Oklad()

{

return Oklad;

}

double Salary::get\_Prize()

{

return Prize;

}

//модификаторы

void Salary::set\_FIO(string N)

{

FIO = N;

}

void Salary::set\_Oklad(int K)

{

Oklad = K;

}

void Salary::set\_Prize(double S)

{

Prize = S;

}

//метод для просмотра атрибутов

void Salary::show()

{

setlocale(LC\_ALL, "Ru");

cout << "Наименование:\t" << FIO << endl;

cout << "Количество:\t" << Oklad << endl;

cout << "Prize:\t" << Prize << endl;

}

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Salary

{

//атрибуты

private:

string FIO;

int Oklad;

double Prize;

public:

Salary();//конструктор без параметров

Salary(string, int, double);//конструктор с параметрами

Salary(const Salary&);//конструктор копирования

~Salary();//деструктор

string get\_FIO();//селектор

void set\_FIO(string);//модификатор

int get\_Oklad();//селектор

void set\_Oklad(int); //модификатор

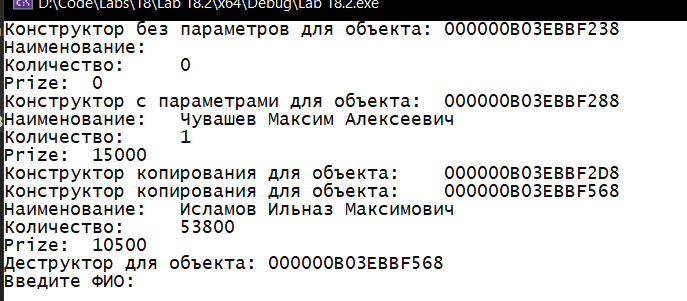
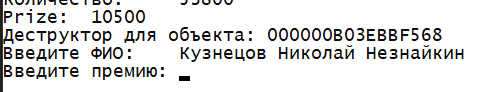
double get\_Prize();//селектор

void set\_Prize(double); //модификатор

void show();//просмотр атрибутов

};

**Скриншоты**

1. 
2. 
3. 