**федеральное государственное автономное образовательное учреждение высшего образования**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)**

**(Факультет информационных технологий)**

***(Институт Принтмедиа и информационных технологий)***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА №6**

**Дисциплина:** Введение в программирование

**Тема: Объекты и классы**

**Выполнил(а): студент(ка) группы 221-377**

Костоваров А.С.

****** (Фамилия И.О.)

**Дата, подпись** 30.09.2022 ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва2022**

**Тема**

**Объекты и классы**

**Цель:** **Получить практические навыки в создании Объекты и классы и их последующем использовании.**

**Код:**

#include <iostream>

#include <cstring>

#include <Windows.h>

using namespace std;

struct task1 {

class Double {

public:

double field;

Double(double v) {

field = v;

}

int method\_zero() {

field = 0;

};

void method\_int(int v) {

field = v;

}

void method\_Enter() {

cout << "Вывод поля на экран: " << field;

}

static Double mothod\_sum(Double v1, Double v2) {

return Double(v1.field + v2.field);

}

};

void task\_main() {

Double v1(0);

Double v2(0);

char space = ' ';

cout << "Введите числа через Enter: \n";

cin >> v1.field;

cin >> v2.field;

Double v3 = Double::mothod\_sum(v1.field, v2.field);

v3.method\_Enter();

}

};

struct task2 {

class Employee {

public:

int nomber;

float salary;

void Input() {

cout << "Введите данные о сотруднике. \n";

cout << "Номер сотрудника:";

cin >> nomber;

cout << "Оклад сотрудника:";

cin >> salary;

}

void Enter() {

cout << "Номер сотрудника в базе: " << nomber << endl;

cout << "Оклад сотрудника в базе: " << salary << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

};

void task\_main() {

Employee mas[3] = { 'v1','v2','v3', };

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

mas[i].Input();

mas[i].Enter();

}

}

};

struct task3 {

class Date {

public:

int month, day, year;

void getdate() {

char slahs = '/';

cout << "Введите время в фромате EU:месяц/день/год.\n";

cin >> month >> slahs >> day >> slahs >> year;

}

void showdate() {

char slahs = '/';

cout << "Вывод времени в формате RU:день/месяц/год.\n";

cout << day << slahs << month << slahs << year;

}

};

void task\_main() {

Date v;

v.getdate();

v.showdate();

}

};

struct task4 {

class Time {

public:

int hour, minute, second;

int day = 0;

Time(int h, int m, int s)

{

hour = h;

minute = m;

second = s;

}

Time()

{

zero(0);

}

void zero(int a = 0) {

hour = a;

minute = a;

second = a;

day = a;

}

void reduction() {

if (second >= 60) {

second = second - 60;

minute = minute + 1;

}

if (minute >= 60) {

minute = minute - 60;

hour = hour + 1;

}

if (hour > 23) {

hour = hour - 24;

day = day + 1;

}

}

void gettime() {

char s = ':';

cout << "Введите время в фромате часы:минуты:секунды\n";

cin >> hour >> s >> minute >> s >> second;

day = 0;

}

static Time sumtime(Time t1, Time t2) {

int hour, minute, second;

hour = t1.hour + t2.hour;

minute = t1.minute + t2.minute;

second = t1.second + t2.second;

return Time(hour, minute, second);

}

void Entertime() {

reduction();

char s = ':';

if (day > 0) {

cout << "Время в базе: " << "Дней: " << day << " | Время: " << hour << s << minute << s << second;

}

else {

cout << "Время в базе: " << hour << s << minute << s << second;

}

}

};

void task\_main() {

Time t1;

t1.gettime();

Time t2;

t2.gettime();

Time t3 = Time::sumtime(t1, t2);

t3.reduction();

t3.Entertime();

}

};

struct task5 {

class Employee {

private:

int nomber;

float salary;

public:

class Date {

public:

int month, day, year;

void getdate() {

char slahs = '/';

cout << "Введите время приема сотрудника на работу в фромате RU(день/месяц/год).\n";

cin >> day >> slahs >> month >> slahs >> year;

}

void showdate() {

char slahs = '/';

cout << "Временя приема сотрудника на работу в формате RU: ";

cout << day << slahs << month << slahs << year;

}

};

class Etype

{

public:

char i;

void etypezero(char d)

{

i = d;

}

void etype()

{

cout << " Введите первую букву должности сотрудника: \n"

"(laborer, secretary, manager, accountant, executive, researcher) : \n";

cin >> i;

}

void print() {

switch (i)

{

case 'l':

cout << "Полное название должности: laborer";

break;

case 's':

cout << "Полное название должности: secretary";

break;

case 'm':

cout << "Полное название должности: manager";

break;

case 'a':

cout << "Полное название должности: accountant";

break;

case 'e':

cout << "Полное название должности: executive";

break;

case 'r':

cout << "Полное название должности: researcher";

break;

}

}

};

Date date;

Etype post;

void getemploy() {

cout << "Введите данные о сотруднике. \n";

cout << "Номер сотрудника:";

cin >> nomber;

cout << "Оклад сотрудника:";

cin >> salary;

date.getdate();

post.etype();

}

void putemploy() {

cout << "Номер сотрудника в базе: " << nomber << endl;

cout << "Оклад сотрудника в базе: " << salary << endl;

date.showdate();

cout << "\n";

post.print();

cout << "\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

};

void task\_main() {

Employee v1;

v1.getemploy();

v1.putemploy();

Employee v2;

v2.getemploy();

v2.putemploy();

Employee v3;

v3.getemploy();

v3.putemploy();

}

};

int main() {

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int checknomber;

for (int checknomber = -1; checknomber != 0;) {

switch (checknomber)

{

case 1:

task1 t1;

t1.task\_main();

cout << "" << endl;

break;

case 2:

task2 t2;

t2.task\_main();

cout << "" << endl;

break;

case 3:

task3 t3;

t3.task\_main();

cout << "" << endl;

break;

case 4:

task4 t4;

t4.task\_main();

cout << "" << endl;

break;

case 5:

task5 t5;

t5.task\_main();

cout << "" << endl;

break;

}

cout << "Для окончания работы введите 0\n";

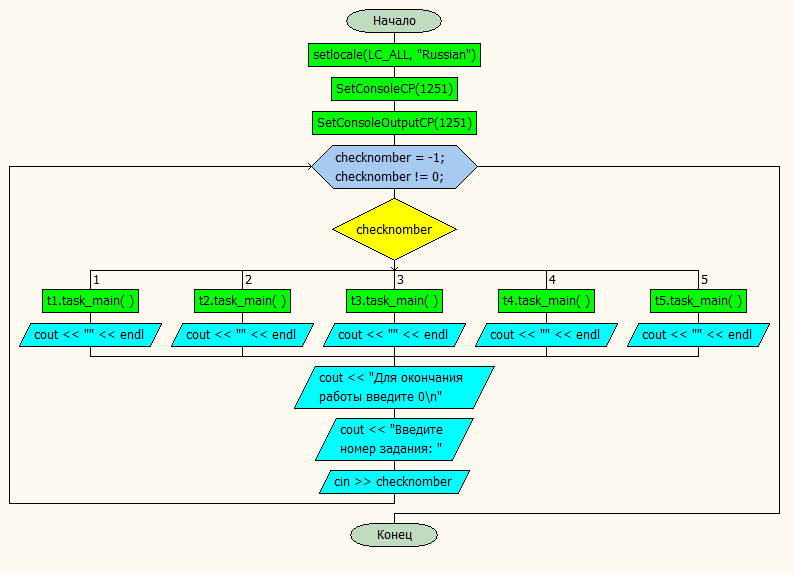
cout << "Введите номер задания: ";

cin >> checknomber;;

}

}

**Блок схема   
main**

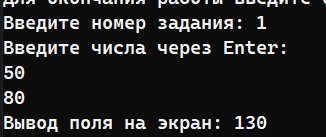
****

**Постановка задачи 1**

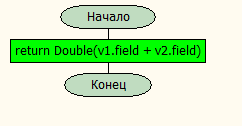
1. Создайте класс Double, имитирующий стандартный тип double. Единственное поле этого класса должно иметь тип double. Создайте методы, которые будут устанавливать значение поля, равным нулю, инициализировать его целым значением, выводить значение поля на экран и складывать два значения типа Double.

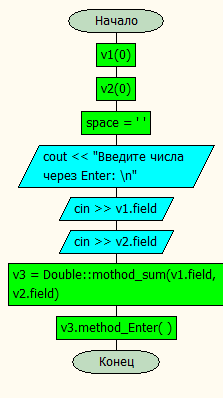
Напишите программу, в которой будут созданы три объекта класса Double, два из которых будут инициализированы. Сложите два инициализированных объекта, присвойте результат третьему, а затем отобразите результат на экране.

**Вывод:**

****

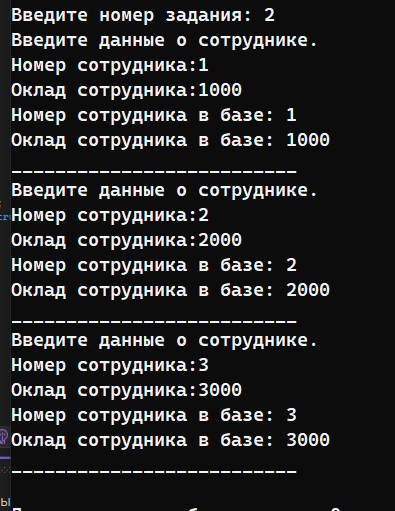
**Блок-схема**

**Task\_main mothod\_sum Класс Duable**

****

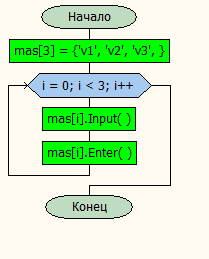
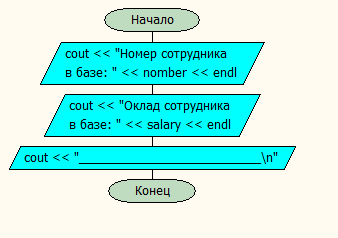
**Постановка задачи 2**

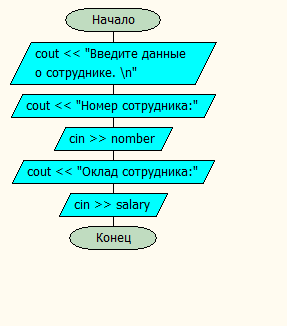
2. Создайте класс **employee.** Класс должен включать поле типа **int** для хранения номера сотрудника и поле типа **float** для хранения величины его оклада. Методы класса должны позволять пользователю вводить и отображать данные класса. Напишите функцию **main(),** которая запросит пользователя ввести данные для трех сотрудников и выведет полученную информацию на экран.

**Вывод:**

**Блок-схема**

**Task\_main Метод enter Класса** Employee

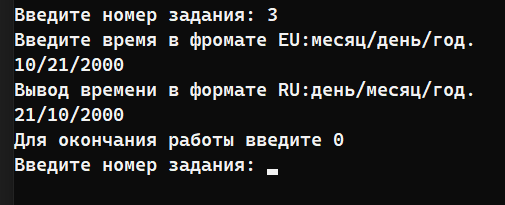
****

**Метод** Input **Класса** Employee ****

**Постановка задачи 3**

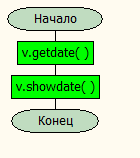
3. Cоздайте класс **date.** Его данные должны размещаться в трех полях типа **int: month, day** и **year.** Метод класса **getdate()** должен принимать значение для объекта в формате 12/31/22, а метод **showdate()** - выводить данные на экран.

**Вывод:**

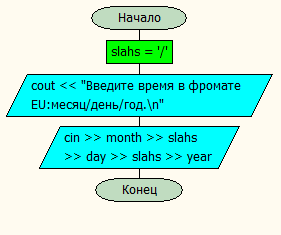
****

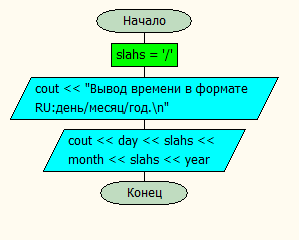
**Блок-схема**

Task\_main



метод getdate класса Date

метод showdate класса Date

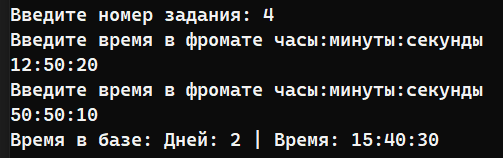


**Постановка задачи 4**

4. Создайте класс с именем time, содержащий три поля типа int, предназначенные для хранения часов, минут и секунд. Один из конструкторов класса должен инициализировать поля нулевыми значениями, а другой конструктор - заданным набором значений. Создайте метод класса, который будет выводить значения полей на экран в формате 11:59:59, и метод, складывающий значения двух объектов типа time, передаваемых в качестве аргументов.

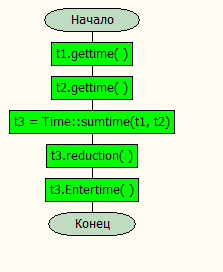
В функции main() следует создать два инициализированных объекта (подумайте, должны ли они быть константными) и один неинициализированный объект. Затем сложите два инициализированных значения, а результат присвойте третьему объекту и выведите его значение на экран. Где возможно, сделайте методы константными.

**Вывод:**

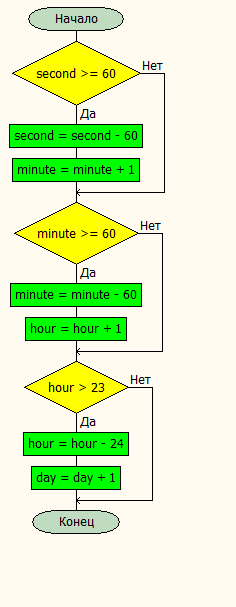
****

**Блок-схема**

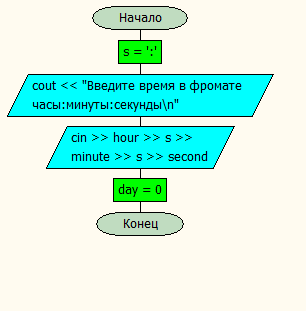
Task\_main

****

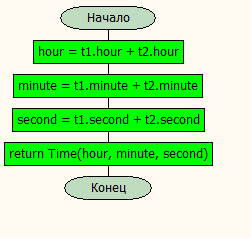
**Метод reduction Класса Time**

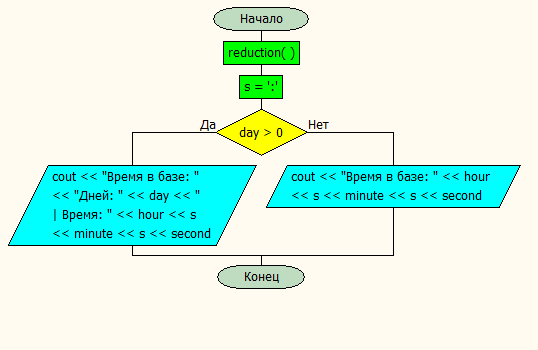


**Метод gettime Класса Time**

****

**Метод sumttime Класса Time**

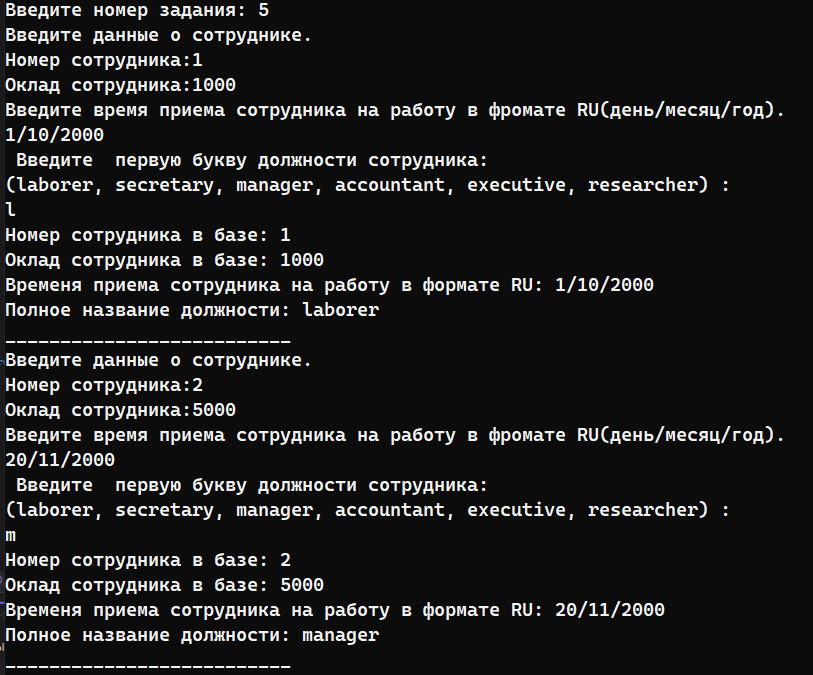
****

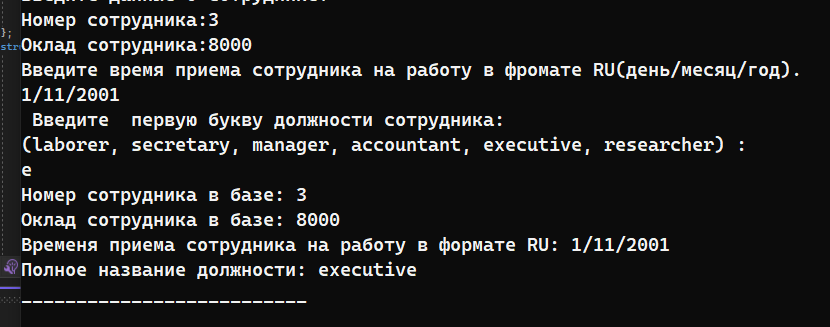
**Метод Enterttime Класса Time**

**Постановка задачи 5**

5. Расширьте содержание класса employee из задачи 2, включив в него класс date и перечисление etype Объект класса date будет использоваться для хранения даты приема сотрудника на работу. Перечисление будет использовано для хранения статуса сотрудника: лаборант, секретарь, менеджер и т. д. Последние два поля данных должны быть закрытыми в определении класса employee, как и номер и оклад сотрудника. Вам будет необходимо разработать методы getemploy() и putemploy(), предназначенные соответственно для ввода и отображения информации о сотруднике. Возможно, при создании методов вам понадобится ветвление switch для работы с перечисляемым типом etype. Напишите функцию main(), которая попросит пользователя ввести данные о трех сотрудниках, а затем выведет эти данные на экран.

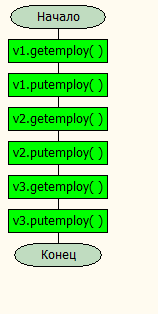
**Вывод:**

****

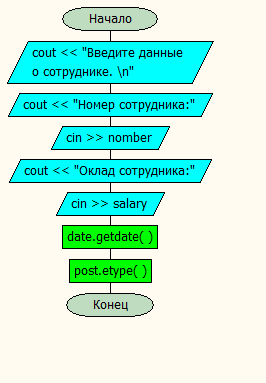
****

**Блок-схема**

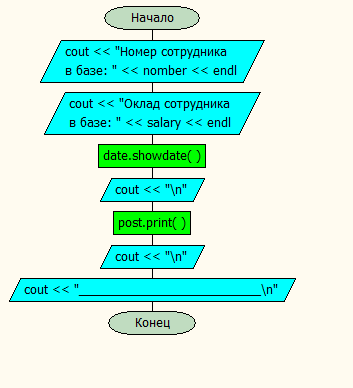
Task\_main

****

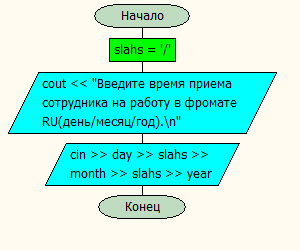
Метод getemploy класса Employee

****

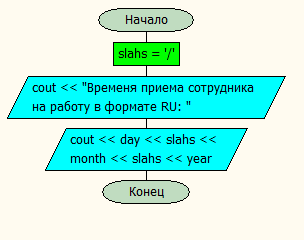
Метод putemploy класса Employee

****

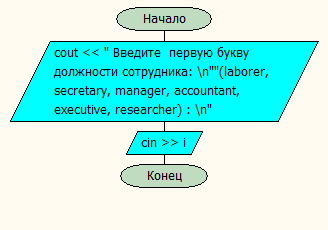
метод getdate класса Date



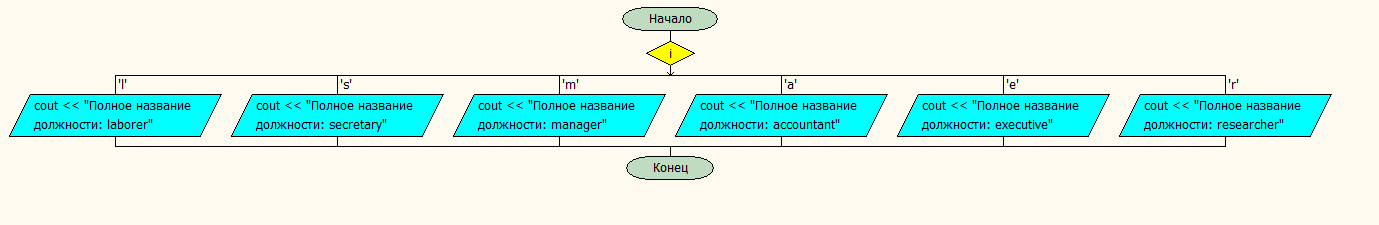
метод showdate класса Date



метод etype класса Etype

****

метод print класса Etype

****

метод print класса Etype

****