Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

ОТЧЕТ

по дисциплине «Практикум по программированию»

**Лабораторная работа № 1**

**Выполнил:**

Cтудент гр. 5130902/40001 Р. В. Белов

**Проверил**

Ст. преподаватель А.М. Журавская

Санкт-Петербург

2025 г.

## **Задание**

1. Выбрать предметную область согласно варианту индивидуального задания.

2. Спроектировать класс для выбранной предметной области.

3. Нарисовать диаграмму спроектированного класса.

4. Предусмотреть наличие у объекта полей, методов и свойств.

5. Предусмотреть наличие свойств только для записи.

Предметная область: **ЖЭС**. В классе хранить информацию о районе, к которому принадлежит ЖЭС, номере ЖЭС, числе жильцов, оплате за месяц (для всех жильцов одна), числе оплативших. Реализовать метод для подсчета общей задолженности жильцов.

Класс *Service* с методами: *totalCost()* – считает общую сумму оплаты ЖЭС в месяц, *currentCost()* – считает текущую сумму платы ЖЭС, учитывая количество жильцов, которые оплатили услуги, *editPrice(price)* – изменяет сумму оплаты ЖЭС в месяц, *editResident(action)* принимает аргумент 0 и 1, 0 – удаляет 1 жильца, оплатившего услуги, 1 – добавляет 1 жильца, оплатившего услуги, *registerHouse()* – зарегестрировать новый дом в системе ЖЭС, *reset()* – количество жильцов, оплативших услуги, приравнивается к нулю.

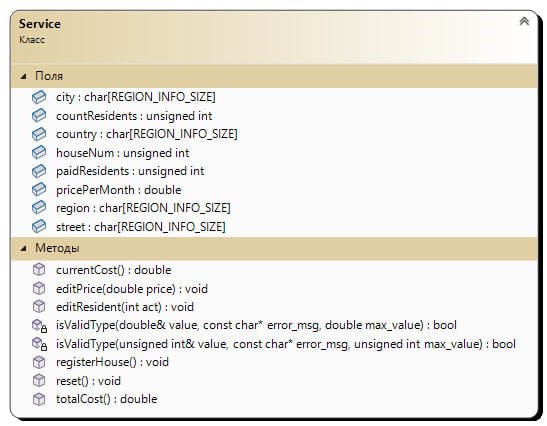


Рисунок 1 – Диаграмма класса Service

## **Код программы**

Код на языке C++:

#include <iostream>

#include <Windows.h>

#include <string>

#include <stdio.h>

#define REGION\_INFO\_SIZE 32

#define MAX\_HOUSE\_NUM 256

#define MAX\_PRICE\_PER\_MONTH 20000

class Service {

public:

char country[REGION\_INFO\_SIZE] = "\0";

char city[REGION\_INFO\_SIZE] = "\0";

char region[REGION\_INFO\_SIZE] = "\0";

char street[REGION\_INFO\_SIZE] = "\0";

unsigned int houseNum;

unsigned int countResidents;

unsigned int paidResidents;

double pricePerMonth;

void registerHouse() {

while (country[0] == '\0') {

std::cout << "Введите страну: ";

scanf\_s("%[^\n]", &country, REGION\_INFO\_SIZE);

if (country[0] == '\0') {

std::cout << "Введите значение!" << std::endl;

std::cin.ignore(256, '\n');

}

}

std::cin.ignore(256, '\n');

while (region[0] == '\0') {

std::cout << "Введите регион: ";

scanf\_s("%[^\n]", &region, REGION\_INFO\_SIZE);

if (region[0] == '\0') {

std::cout << "Введите значение!" << std::endl;

std::cin.ignore(256, '\n');

}

}

std::cin.ignore(256, '\n');

while (city[0] == '\0') {

std::cout << "Введите город: ";

scanf\_s("%[^\n]", &city, REGION\_INFO\_SIZE);

if (city[0] == '\0') {

std::cout << "Введите значение!" << std::endl;

std::cin.ignore(256, '\n');

}

}

std::cin.ignore(256, '\n');

while (street[0] == '\0') {

std::cout << "Введите улицу: ";

scanf\_s("%[^\n]", &street, REGION\_INFO\_SIZE);

if (street[0] == '\0') {

std::cout << "Введите значение!" << std::endl;

std::cin.ignore(256, '\n');

}

}

std::cin.ignore(256, '\n');

std::cout << "Введите номер дома: ";

while (!isValidType(houseNum, "Номер дома должен быть целым числом!\n", MAX\_HOUSE\_NUM)) {

std::cout << "Введите номер дома: ";

};

std::cout << "Введите количество жильцов в доме: ";

while (!isValidType(countResidents, "Число жильцов должно быть натуральным!\n", (int)1e+6)) {

std::cout << "Введите количество жильцов в доме: ";

}

std::cout << "Введите сумму оплаты за ЖЭС в месяц: ";

while (!isValidType(pricePerMonth, "Сумма ЖЭС должна быть вещественным числом!\n", MAX\_PRICE\_PER\_MONTH)) {

std::cout << "Введите сумму оплаты за ЖЭС в месяц: ";

}

std::cout << "Дом по адресу: регион " << region << ", г. " << city << ", ул. " << street << ", д. " << houseNum << " зарегестрирован в системе!" << std::endl;

printf\_s("Число жильцов: %d чел.\nУстановленная плата: %.2f р/мес\nОбщая сумма платы ЖЭС: %.2f р/мес\n", countResidents, pricePerMonth, totalCost());

}

double totalCost() {

return countResidents \* pricePerMonth;

}

double currentCost() {

return totalCost() - (paidResidents \* pricePerMonth);

}

void editResident(int act) {

switch (act) {

case 1:

countResidents++;

case 0:

countResidents--;

default:

countResidents++;

}

}

void editPrice(double price) {

pricePerMonth = price;

}

void reset() {

paidResidents = 0;

}

private:

bool isValidType(double& value, const char\* error\_msg = "Введён неверный тип данных!\n", double max\_value = MAX\_PRICE\_PER\_MONTH) {

char c;

char result[16];

size\_t index = 0;

while (std::cin.get(c)) {

if (c == '\n') break;

if (isdigit(c) || c == '.') {

result[index] = c;

index++;

}

else {

std::cout << error\_msg;

std::cin.ignore(64, '\n');

return false;

}

}

result[index] = '\0';

value = std::stod(result);

if (value > max\_value) {

std::cout << "Превышено максимальное число! (" << max\_value << ")" << std::endl;

return false;

}

std::cin.clear();

return true;

}

bool isValidType(unsigned int& value, const char\* error\_msg = "Введён неверный тип данных!\n", unsigned int max\_value = 1e+6) {

char c;

char result[16];

size\_t index = 0;

while (std::cin.get(c)) {

if (c == '\n') break;

if (isdigit(c)) {

result[index] = c;

index++;

}

else {

std::cout << error\_msg;

std::cin.ignore(64, '\n');

return false;

}

}

result[index] = '\0';

value = std::atoi(result);

if (value > max\_value) {

std::cout << "Превышено максимальное число! (" << max\_value << ")" << std::endl;

return false;

}

std::cin.clear();

return true;

}

};

int main() {

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

Service service;

service.registerHouse();

}

Код на языке Python:

MAX\_INPUT\_SIZE = 256

MAX\_HOUSE\_NUM = 256

MAX\_PRICE\_PER\_MONTH = 20\_000

class Service:

def \_\_init\_\_(self):

self.country = None

self.city = None

self.region = None

self.street = None

self.houseNum = None

self.countResidents = None

self.paidResidents = None

self.pricePerMonth = 0

self.registerHouse()

def registerHouse(self):

"""

Регистрация нового дома в системе

"""

while not self.country:

try:

self.country = input("Введите страну: ")

if len(self.country) > MAX\_INPUT\_SIZE:

print(f"Превышно максимальное количество символов! ({MAX\_INPUT\_SIZE})")

self.country = None

if len(self.country) == 0:

print(f"Введите значение!")

self.country = None

except:

print("Произошла ошибка при вводе данных!")

while not self.region:

try:

self.region = input("Введите регион: ")

if len(self.region) > MAX\_INPUT\_SIZE:

print(f"Превышно максимальное количество символов! ({MAX\_INPUT\_SIZE})")

self.region = None

if len(self.region) == 0:

print(f"Введите значение!")

self.region = None

except:

print("Произошла ошибка при вводе данных!")

while not self.city:

try:

self.city = input("Введите город: ")

if len(self.city) > MAX\_INPUT\_SIZE:

print(f"Превышно максимальное количество символов! ({MAX\_INPUT\_SIZE})")

self.city = None

if len(self.city) == 0:

print(f"Введите значение!")

self.city = None

except:

print("Произошла ошибка при вводе данных!")

while not self.street:

try:

self.street = input("Введите улицу: ")

if len(self.street) > MAX\_INPUT\_SIZE:

print(f"Превышно максимальное количество символов! ({MAX\_INPUT\_SIZE})")

self.street = None

if len(self.street) == 0:

print(f"Введите значение!")

self.street = None

except:

print("Произошла ошибка при вводе данных!")

while not self.houseNum:

try:

self.houseNum = int(input("Введите номер дома: "))

if self.houseNum > MAX\_HOUSE\_NUM:

print(f"Превышно максимальное число! ({MAX\_HOUSE\_NUM})")

self.houseNum = None

if self.houseNum == 0:

print(f"Введите значение!")

self.houseNum = None

except:

print("Произошла ошибка при вводе данных!")

while not self.countResidents:

try:

self.countResidents = int(input("Введите число жителей дома: "))

if self.countResidents > 1e+6:

print(f"Превышно максимальное число! ({1e+6})")

self.countResidents = None

if self.countResidents == 0:

print(f"Введите значение!")

self.countResidents = None

except:

print("Произошла ошибка при вводе данных!")

while not self.pricePerMonth:

try:

self.pricePerMonth = float(input("Введите плату за услуги ЖЭС в месяц: "))

if self.pricePerMonth > MAX\_PRICE\_PER\_MONTH:

print(f"Превышно максимальное число! ({MAX\_PRICE\_PER\_MONTH})")

self.pricePerMonth = None

if self.pricePerMonth == 0:

print(f"Введите значение!")

self.pricePerMonth = None

except:

print("Произошла ошибка при вводе данных!")

print(f"Зарегестрирован дом по адресу: страна {self.city}, рег. {self.region}, г. {self.city}, ул. {self.street}, д. {self.houseNum}")

print(f"Число жильцов: {self.countResidents}")

print(f"Установленная плата: {self.pricePerMonth} р/мес.")

print(f"Общая сумма платы ЖЭС: {self.totalCost()} р/мес.")

def totalCost(self) -> float:

"""

Подсчёт общей суммы платы за услуги ЖЭС

"""

return self.countResidents \* self.pricePerMonth

def currentCost(self) -> float:

"""

Подсчет текущей платы за услуги ЖЭС

"""

return self.totalCost() - (self.paidResidents \* self.pricePerMonth)

def editResident(self, act) -> None:

if act == 1:

self.paidResidents += 1

return

if act == 0:

self.paidResidents -= 1

return

self.paidResidents += 1

def reset(self):

self.paidResidents = 0

def editPrice(self, price):

self.pricePerMonth = price

a = Service()

**Пример работы программы**

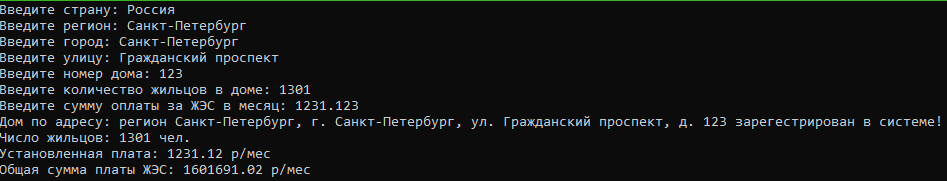


Рисунок 2 – Правильная работа программы на C++

## **Вывод**

В ходе лабораторной работы, была написана программа на языках программирования C++ и Python. Был спроектирован класс Service, затем были написаны методы для обращения к нему. Также была нарисована диаграмма спроектированного класса, которая имеет методы класса и типы данных полей.

Благодаря этой лабораторной работе, я научился создавать и применять классы в коде. Объектно-ориентированное программирование будет часто использоваться в крупных, более сложных проектах.