



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

Отчет

по лабораторной работе № 7

Название лабораторной работы: Простые объекты. Конструкторы.

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-14Б

18.11.2023

А.С. Воеводин

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

18.11.2023

О.А. Веселовская

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Цель работы – Решить поставленную задачу, используя простые объекты и их конструкторы.

Задание – Построить диаграмму класса для реализации описанного ниже объекта. Составить программный код и написать тестирующую программу.

Все поля класса должны быть частными (private) или защищенными (protected).

Методы не должны содержать операций ввода/вывода, за исключением процедуры, единственной задачей которой является вывод информации об объекте на экран.

Объект – лифт. Поле: номер этажа, на котором в данный момент находится лифт.

Методы: конструктор, процедура перемещения лифта на заданный этаж и функция, возвращающая номер текущего этажа.

Ход работы:

- Написание программы.
- Тестирование программы при различных тестовых данных.
- Изображение схемы алгоритмов программы и диаграммы классов.
- Вывод.

Для начала создадим многофайловую структуру, а именно: файлы Elevator.hpp, Elevator.cpp, что позволит удобно работать с классами. Далее напишем код программы:

```
#pragma once

class Elevator
{
public:
    Elevator();
    Elevator(const long long int);
    ~Elevator();

    void SetFloor(const long long int);
    long long int GetFloor();
    void PrintInfo();

private:
    long long int floor;
};
```

Рисунок 1 – Объявление класса Elevator, его полей, конструкторов и методов

```
#include "Elevator.hpp"
#include <iostream>

Elevator::Elevator() : floor(0) {}
Elevator::Elevator(const long long floor_) : floor(floor_) {}
Elevator::~~Elevator() { floor = 0; }
void Elevator::SetFloor(const long long floor_) { floor = floor_; }
long long int Elevator::GetFloor() { return floor; }
void Elevator::PrintInfo() {
    std::cout << "Your elevator on " << floor << " floor" << std::endl << std::endl;
}
```

Рисунок 2 – Реализация класса Elevator, его полей, конструкторов и методов

```

#include "Elevator.hpp"
#include <iostream>
#include <string>

bool isDigit(const std::string& tempStr) {
    bool isAllDigit = false;
    if (!tempStr.empty() && std::isdigit(tempStr[0]) || tempStr[0] == '-') {
        isAllDigit = tempStr.length() > 1 || std::isdigit(tempStr[0]);
        for (size_t i = 1; i < tempStr.size(); ++i) {
            if (!std::isdigit(tempStr[i])) {
                isAllDigit = false;
            }
        }
    }

    return isAllDigit;
}

int main()
{
    bool isBreak = false;
    char userChoice = '\0';
    bool isCreatedElevator = false;
    Elevator elevator;
    while (!isBreak) {
        std::cout << "Choose command:" << std::endl
        << "1) Create new elevator" << std::endl
        << "2) Set floor for your elevator" << std::endl
        << "3) Print info about your elevator" << std::endl
        << "4) End" << std::endl << std::endl;
        std::cin >> userChoice;

        switch (userChoice)
        {
            case '1': {

```

Рисунок 3 – Основная программа часть 1

```

switch (userChoice)
{
case '1': {
    std::string ans;
    std::cout << "Enter your floor for elevator: ";
    std::cin >> ans;
    while (!isDigit(tempStr: ans)) {
        std::cout << "Wrong characters. Only digits. Enter again: ";
        std::cin >> ans;
    }
    std::cout << std::endl;
    elevator.SetFloor(std::stoi(str: ans));
    isCreatedElevator = true;
    break;
}
case '2': {
    if (!isCreatedElevator) {
        std::cout << "Create your elevator first." << std::endl << std::endl;
    }
    else {
        std::string ans;
        std::cout << "Enter your floor for elevator: ";
        std::cin >> ans;
        while (!isDigit(tempStr: ans)) {
            std::cout << "Wrong characters. Only digits. Enter again: ";
            std::cin >> ans;
        }
        elevator.SetFloor(std::stoi(str: ans));
    }
    std::cout << std::endl;
    break;
}
}

```

Рисунок 4 – Основная программа часть 2

```

        case '3': {
            if (isCreatedElevator) {
                elevator.PrintInfo();
            }
            else {
                std::cout << "Create your elevator first." << std::endl << std::endl;
            }
            break;
        }
        case '4': {
            isBreak = true;
            break;
        }
        default:
            std::cout << "Wrong command. Try again" << std::endl << std::endl;
            break;
    }
}
return 0;
}

```

Рисунок 5 – Основная программа часть 3

```

Choose command:
1) Create new elevator
2) Set floor for your elevator
3) Print info about your elevator
4) End

2
Create your elevator first.

Choose command:
1) Create new elevator
2) Set floor for your elevator
3) Print info about your elevator
4) End

3
Create your elevator first.

Choose command:
1) Create new elevator
2) Set floor for your elevator
3) Print info about your elevator
4) End

4

C:\Users\artem\source\repos\laba_7\x64\Debug\laba_7.exe (процесс 8592) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:|

```

Рисунок 6 – Тестовые команды часть 1

```
Choose command:
1) Create new elevator
2) Set floor for your elevator
3) Print info about your elevator
4) End

1
Enter your floor for elevator: 10

Choose command:
1) Create new elevator
2) Set floor for your elevator
3) Print info about your elevator
4) End

2
Enter your floor for elevator: 12

Choose command:
1) Create new elevator
2) Set floor for your elevator
3) Print info about your elevator
4) End

3
Your elevator on 12 floor

Choose command:
1) Create new elevator
2) Set floor for your elevator
3) Print info about your elevator
4) End

2
Enter your floor for elevator: 1

Choose command:
1) Create new elevator
2) Set floor for your elevator
3) Print info about your elevator
4) End

3
Your elevator on 1 floor
```

Рисунок 7 – Тестовые команды часть 2

Можем убедиться, что программа работает верно. Далее изобразим алгоритм в виде схемы:

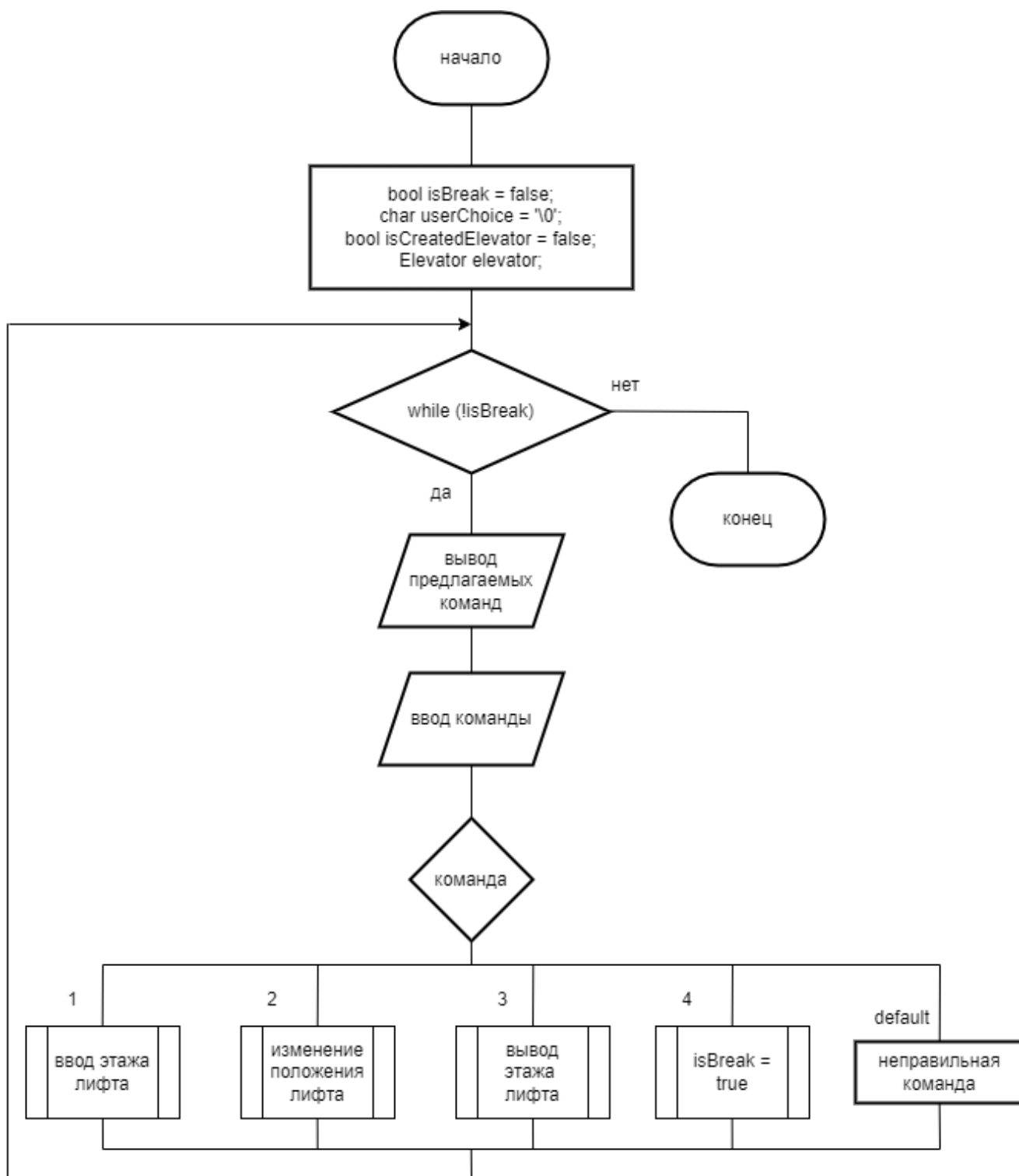


Рисунок 8 – Схема алгоритма программы

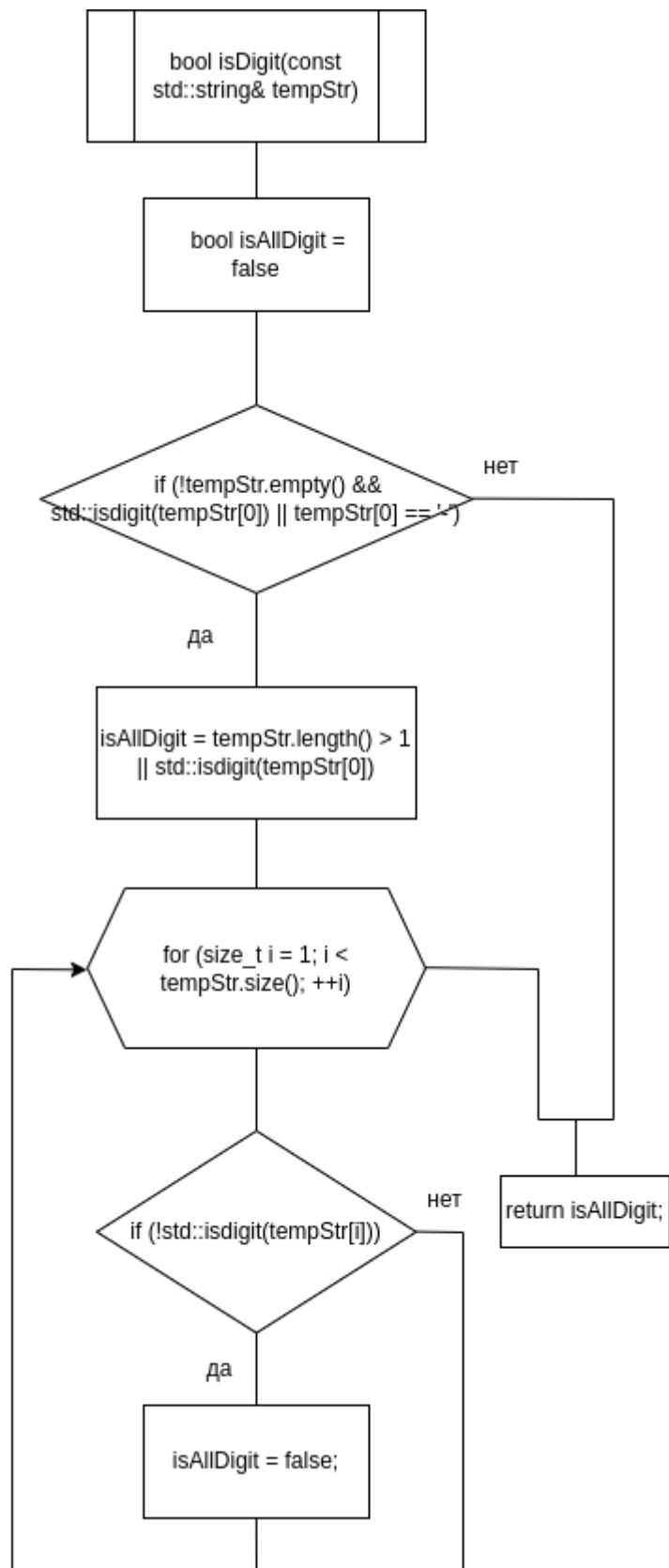


Рисунок 9 – Функция, используемая в программе

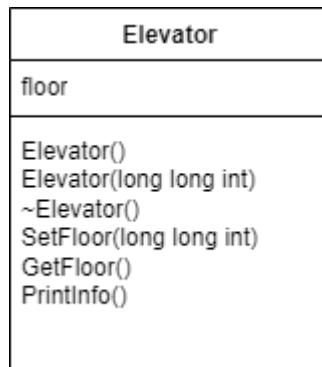


Рисунок 10 – Диаграмма класса Elevator

Вывод: В ходе лабораторной работы были получены навыки работы с классами, взаимодействия с пользователями, написание и изображение классов в программе и в виде схемы соответственно. А также работа с конструкторами.