



**«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 домашнему заданию № 4

Название домашнего задания: Основы объектно-ориентированного программирования.

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-14Б

02.12.2023

А.С. Воеводин

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

02.12.2023

О.А. Веселовская

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Цель задания

Работа с объектно-ориентированным программированием.

Задание

Разработать класс для реализации указанного объекта. Построить его диаграмму и составить программный код.

Объект – атом химического элемента. Поля: символьное обозначение химического элемента (например, Hg или S) и его относительная атомная масса. Методы: конструктор, процедура вывода на экран информации об атоме и функции, возвращающие значения полей по запросу.

Часть 1. Массив объектов

Разработать тестирующую программу, в которой описана совокупность объектов, определяющих молекулу. Для молекулы на экран должна выводиться формула химического вещества в виде Fe_2O_3 и вычисляться ее относительная масса.

Часть 2. Композиция

Используя разработанный в первой части задания класс, построить диаграмму классов композиционного объекта. Составить программный код описания класса. Разработать тестирующую программу.

Объект – молекула химического вещества, состоящая из некоторого количества атомов. Методы объекта должны позволять: сконструировать объект, вывести на экран формулу химического вещества в виде Fe_2O_3 , вычислить относительную молекулярную массу.

Ход работы:

- Написание программы и схемы алгоритма в части 1
- Тестировка программы в части 1
- Написание программы и схемы алгоритма в части 2
- Тестировка программы в части 2
- Вывод

Для начала создадим заголовочный файл Atom.hpp и файл реализации Atom.cpp:

```
#ifndef DZ_4__ATOM_HPP_
#define DZ_4__ATOM_HPP_

#include <string>
#include <unordered_map>

class Atom {
public:
    Atom();
    Atom(std::string);
    void PrintInfo();
    std::string GetName();
    double GetMass();
    const static std::unordered_map<std::string, double> mendeleev;

private:
    std::string name;
    double mass;
};

#endif //DZ_4__ATOM_HPP_
```

Рисунок 1 – Заголовочный файл Atom.hpp

```

    { x: "Si", y: 28.086},
    { x: "Sm", y: 150.36},
    { x: "Sn", y: 118.710},
    { x: "Sr", y: 87.62},
    { x: "Ta", y: 180.948},
    { x: "Tb", y: 158.925},
    { x: "Tc", y: 97.907},
    { x: "Te", y: 127.60},
    { x: "Th", y: 232.038},
    { x: "Ti", y: 47.88},
    { x: "Tl", y: 204.383},
    { x: "Tm", y: 168.934},
    { x: "U", y: 238.029},
    { x: "V", y: 50.942},
    { x: "W", y: 183.85},
    { x: "Xe", y: 131.29},
    { x: "Y", y: 88.906},
    { x: "Yb", y: 173.04},
    { x: "Zn", y: 65.39},
    { x: "Zr", y: 91.224}
};

Atom::Atom() : mass(0.0) {}
Atom::Atom(std::string name_) : name(name_), mass(mendelev.at( k: name_)) {}
void Atom::PrintInfo() {
    std::cout << "name: " << name << " atomic mass: " << mass << std::endl;
}
std::string Atom::GetName() {
    return name;
}
double Atom::GetMass() {
    return mass;
}

```

Рисунок 2 – Файл реализации Atom.cpp

```

#include "Atom.hpp"
#include <vector>
#include <iostream>

int main() {
    std::vector<Atom> atoms;
    std::string nameOfMolecula;
    std::string tempNameOfAtom;
    int countOfAtom = 0;
    double massOfMolecula = 0.0;

    while (tempNameOfAtom != "end") {
        std::cout << "Enter your atom and amount of him or end: ";
        std::cin >> tempNameOfAtom;
        if (tempNameOfAtom != "end") {
            std::cin >> countOfAtom;
            if (countOfAtom <= 0 || Atom::mendeLeev.find(x: tempNameOfAtom) == Atom::mendeLeev.end()) {
                std::cout << "Try again. Wrong name or non natural number" << std::endl;
            } else {
                atoms.emplace_back<Atom>(tempNameOfAtom);
                nameOfMolecula += (countOfAtom > 1) ? tempNameOfAtom + std::to_string(val: countOfAtom) : tempNameOfAtom;
                massOfMolecula += Atom::mendeLeev.at(k: tempNameOfAtom) * countOfAtom;
            }
        }
    }

    std::cout << nameOfMolecula << ' ' << massOfMolecula;

    return 0;
}

```

Рисунок 3 – Основная программа

Теперь введём несколько атомов, составляющих молекулу:

```

Enter your atom and amount of him or end: Fe 2
Enter your atom and amount of him or end: O 3
Enter your atom and amount of him or end: end
Fe2O3 159.691
Process finished with exit code 0

```

Рисунок 4 – Тестовые данные часть 1

```

Enter your atom and amount of him or end: H 2
Enter your atom and amount of him or end: O 1
Enter your atom and amount of him or end: end
H2O 18.015
Process finished with exit code 0

```

Рисунок 5 – Тестовые данные часть 2

```
Enter your atom and amount of him or end: H 2
Enter your atom and amount of him or end: S 1
Enter your atom and amount of him or end: O 4
Enter your atom and amount of him or end: end
H2SO4 98.078
Process finished with exit code 0
```

Рисунок 6 – Тестовые данные часть 3

Можно проверить вычисления на специализированных сайтах и убедиться, что программа работает верно. Теперь изобразим программу в виде схемы и диаграммы:

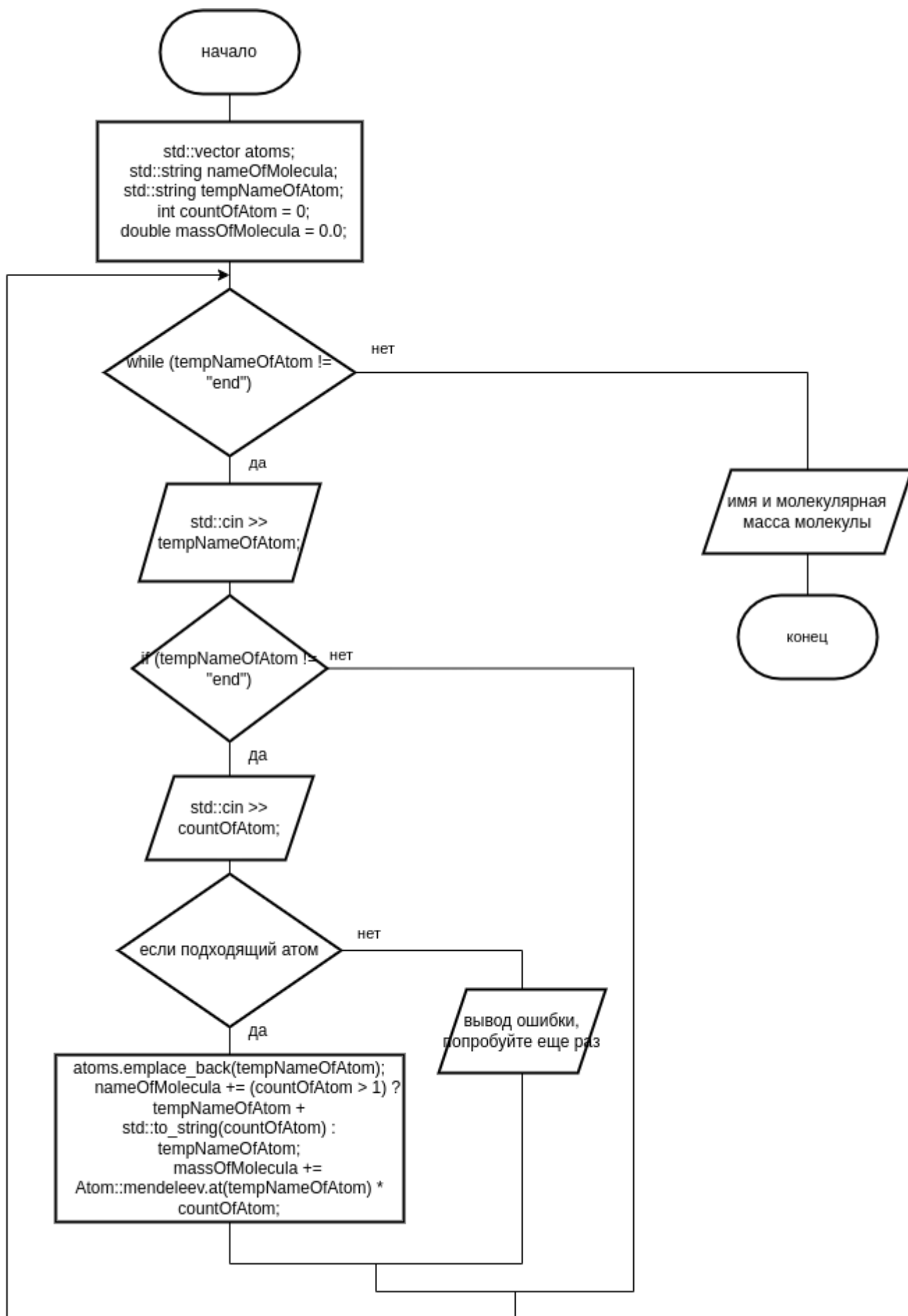


Рисунок 7 – Схема алгоритма основной программы

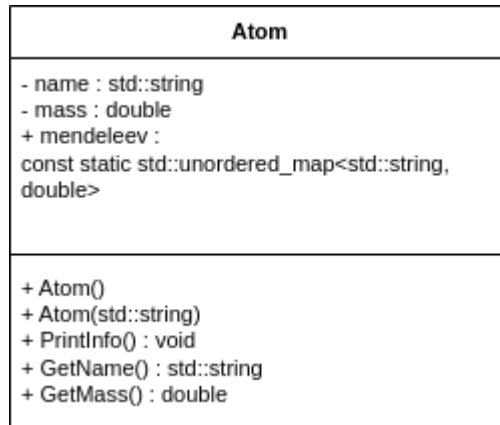


Рисунок 8 – Диаграмма класса Atom

Теперь допишем программу для второй части задания:

```
std::string tempNameOfAtom;
int countOfAtom = 0;
Molecule molecule;

while (tempNameOfAtom != "end") {
    std::cout << "Enter your atom and amount of him or end: ";
    std::cin >> tempNameOfAtom;
    if (tempNameOfAtom != "end") {
        std::cin >> countOfAtom;
        if (countOfAtom >= 1 && Molecule::CheckAtom(tempNameOfAtom)) {
            molecule.AddAtom(tempNameOfAtom, countOfAtom);
        } else {
            std::cout << "Try again. Wrong name or non natural number" << std::endl;
        }
    }
}
molecule.PrintInfo();
std::cout << molecule.CalcMass() << std::endl;
return 0;
```

Рисунок 9 – Основная программа


```
#ifndef DZ_4__MOLECULE_HPP_  
#define DZ_4__MOLECULE_HPP_  
  
#include <vector>  
#include <string>  
  
class Molecule {  
public:  
    void AddAtom(const std::string&, size_t);  
    void PrintInfo();  
    double CalcMass();  
    static bool CheckAtom(const std::string&);  
  
private:  
    class Atom;  
    std::vector<std::pair<Atom, size_t>> atoms;  
};  
  
#endif //DZ_4__MOLECULE_HPP_
```

Рисунок 10 – Заголовочный файл Molecule.hpp

```

#include "Molecule.hpp"
#include "Atom.hpp"
#include <utility>
#include <iostream>

void Molecule::AddAtom(const std::string& name_, size_t count) {
    atoms.emplace_back(std::move(std::make_pair(Molecule::Atom(name_), [>>]count)));
}

void Molecule::PrintInfo() {
    for (auto& atom:pair<Atom, size_t>& : atoms) {
        std::cout << ((atom.second > 1) ? atom.first.GetName() + std::to_string(atom.second) : atom.first.GetName());
    }
    std::cout << std::endl;
}

double Molecule::CalcMass() {
    double mass = 0.0;
    for (auto& atom:pair<Atom, size_t>& : atoms) {
        mass += atom.first.GetMass() * atom.second;
    }
    return mass;
}

bool Molecule::CheckAtom(const std::string& name_) {
    return Molecule::Atom::mendelev.find(name_) != Molecule::Atom::mendelev.end();
}

```

Рисунок 11 – Файл реализации Molecule.cpp

Теперь введем те же тестовые данные для проверки корректности результата:

```

Enter your atom and amount of him or end: Fe 2
Enter your atom and amount of him or end: 0 3
Enter your atom and amount of him or end: end
Fe203
159.691

```

Рисунок 12 – Тестовые данные часть 1

```

Enter your atom and amount of him or end: H 2
Enter your atom and amount of him or end: 0 1
Enter your atom and amount of him or end: end
H20
18.015

```

Рисунок 13 – Тестовые данные часть 2

```
Enter your atom and amount of him or end: H 2
Enter your atom and amount of him or end: S 1
Enter your atom and amount of him or end: O 4
Enter your atom and amount of him or end: end
H2SO4
98.078
```

Рисунок 14 – Тестовые данные часть 3

Как видно из рисунков, программа отработала верно. Теперь изобразим программу в виде схемы и диаграммы:

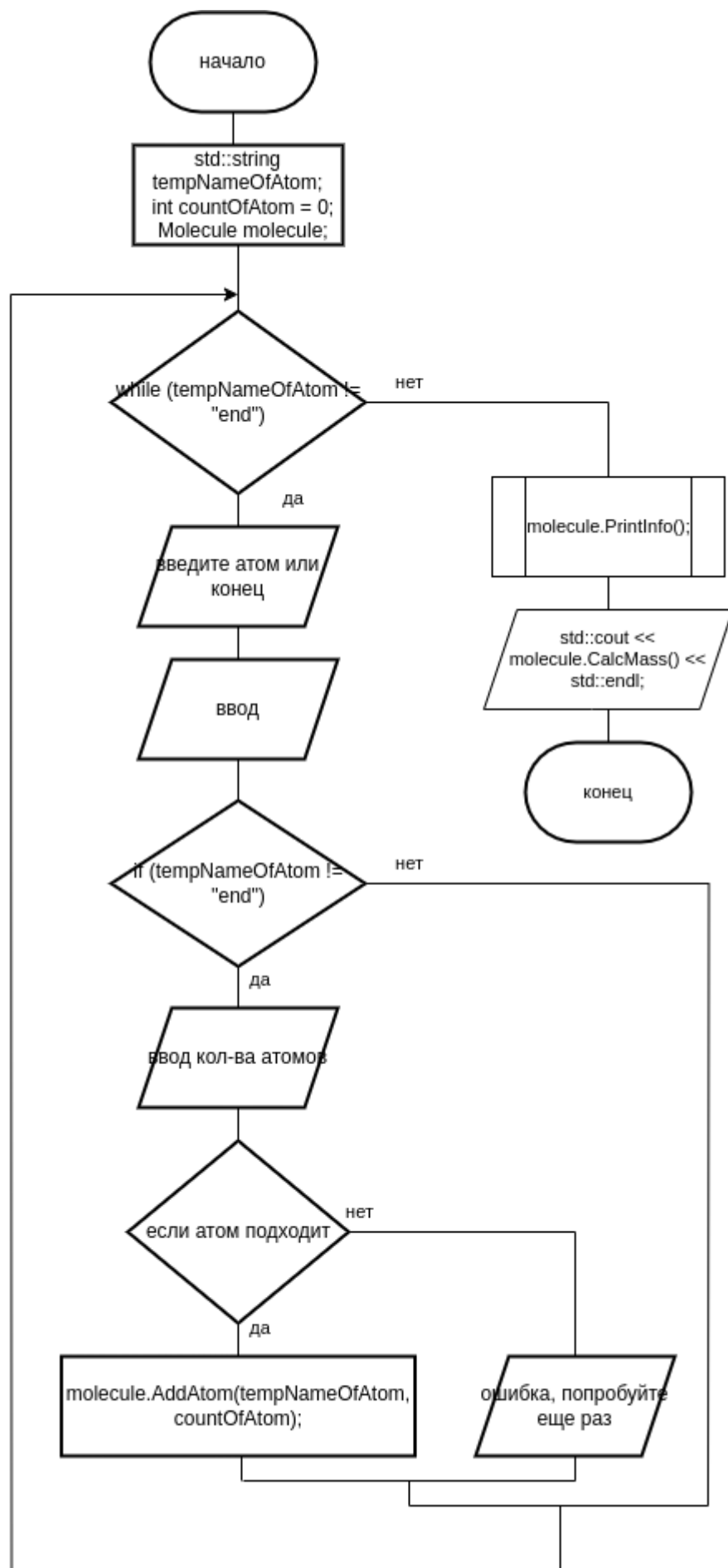


Рисунок 15 – Схема алгоритма основной программы

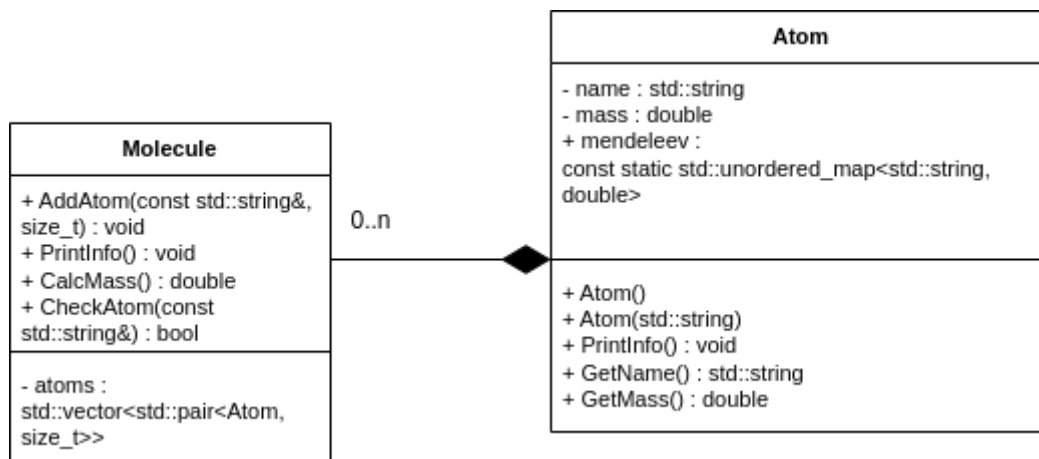


Рисунок 16 – Диаграмма классов

Вывод

В ходе домашнего задания были получены навыки работы с одним из основополагающих способов программирования в стиле ООП – композицией, также навык работы с UML-диаграммами, позволяющие описать отношения классов в виде схемы.