

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.01 – «Информатика и вычислительная
техника»

**Отчет по
лабораторной работе 7 (классы)**

Выполнил студент гр. ИВТ-23-16
Давыдов Андрей Юрьевич

Проверил:

ст. преп. каф. ИТАС

Яруллин Денис Владимирович

(оценка)

(подпись)

(дата)

г. Пермь, 2023

Постановка задачи

Класс- контейнер СПИСОК с ключевыми значениями типа `int`.

Реализовать операции:

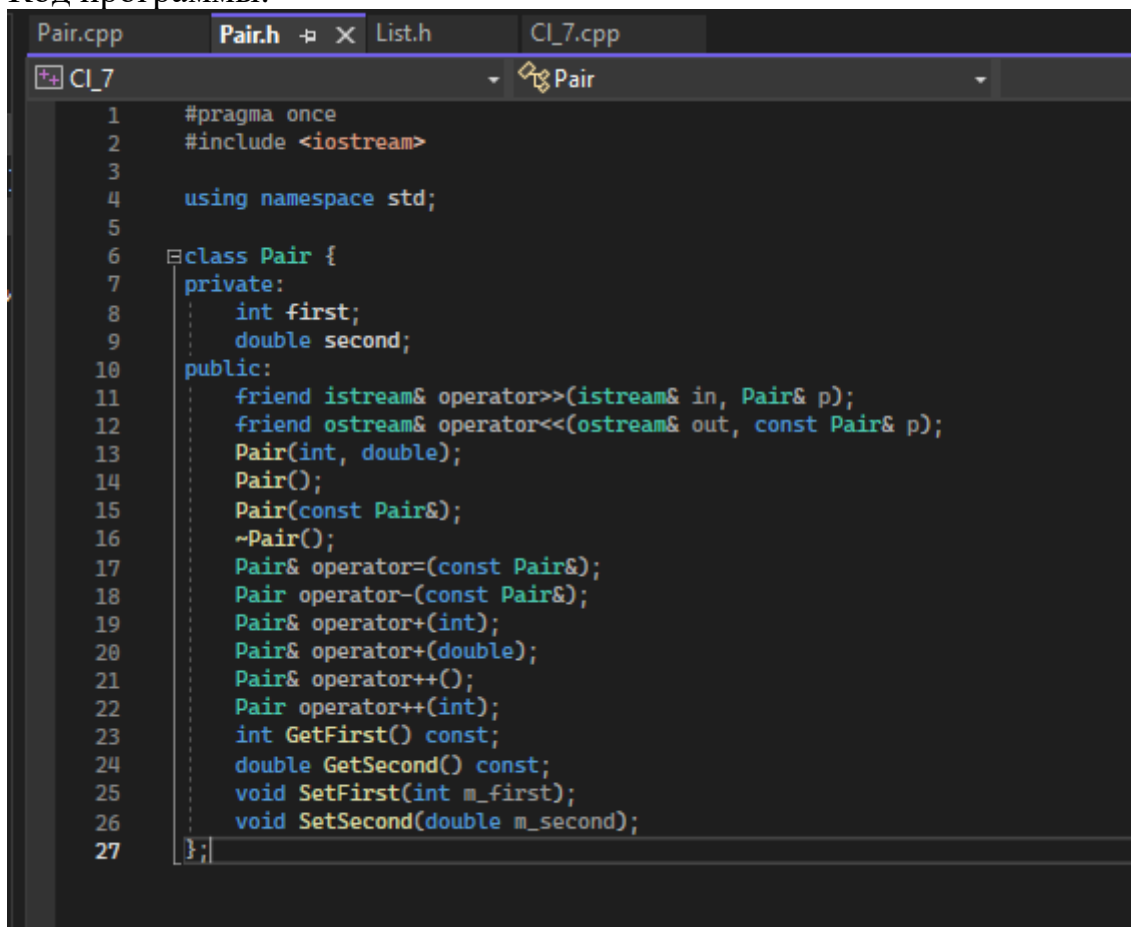
`[]` – доступа по индексу;

`int()` – определение размера списка;

* вектор – умножение элементов списков `a[i]*b[i]`;

Пользовательский класс `Pair` (пара чисел). Пара должна быть представлено двумя полями: типа `int` для первого числа и типа `double` для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием.

Код программы:



```
Pair.cpp  Pair.h  List.h  Cl_7.cpp
Cl_7
Pair
1  #pragma once
2  #include <iostream>
3
4  using namespace std;
5
6  class Pair {
7  private:
8      int first;
9      double second;
10 public:
11     friend ostream& operator>>(ostream& out, const Pair& p);
12     friend istream& operator<<(istream& in, Pair& p);
13     Pair(int, double);
14     Pair();
15     Pair(const Pair&);
16     ~Pair();
17     Pair& operator=(const Pair&);
18     Pair operator-(const Pair&);
19     Pair& operator+(int);
20     Pair& operator+(double);
21     Pair& operator++();
22     Pair operator++(int);
23     int GetFirst() const;
24     double GetSecond() const;
25     void SetFirst(int m_first);
26     void SetSecond(double m_second);
27 }
```

```

1
2     #include "Pair.h"
3
4     Pair::Pair(int first, double second) {
5         this->first = first;
6         this->second = second;
7     }
8     istream& operator>>(istream& in, Pair& m_pair) {
9         in >> m_pair.first;
10        in >> m_pair.second;
11        return in;
12    }
13    ostream& operator<<(ostream& out, const Pair& m_pair) {
14        return (out << m_pair.first << " : " << m_pair.second);
15    }
16    Pair::Pair() {
17        first = 0;
18        second = 0;
19    }
20    Pair::~~Pair() {
21    }
22    Pair::Pair(const Pair& m_pair) {
23        first = m_pair.first;
24        second = m_pair.second;
25    }
26    Pair& Pair::operator=(const Pair& m_pair) {
27        if (&m_pair != this) {
28            first = m_pair.first;
29            second = m_pair.second;
30        }
31        return *this;
32    }
33    Pair Pair::operator-(const Pair& m_pair) {
34        Pair result(first - m_pair.first, second - m_pair.second);
35        return result;
36    }
37    }
38    Pair& Pair::operator+(int first) {
39        this->first += first;
40        return *this;
41    }
42    Pair& Pair::operator+(double second) {
43        this->second += second;
44        return *this;
45    }
46    Pair& Pair::operator++() {
47        ++first;
48        ++second;
49        return *this;
50    }
51    Pair Pair::operator ++(int) {
52        Pair temp = *this;
53        this->first++;
54        this->second++;
55        return temp;
56    }

```

```
Pair.cpp  Pair.h  List.h  CL_7.cpp
CL_7
Pair
58 int Pair::GetFirst() const {
59     return first;
60 }
61
62 double Pair::GetSecond() const {
63     return second;
64 }
65
66 void Pair::SetFirst(int m_first) {
67     first = m_first;
68 }
69
70 void Pair::SetSecond(double m_second) {
71     second = m_second;
72 }
73
```

```
Pair.cpp  Pair.h  List.h  Cl_7.cpp
Cl_7  Node<T>
1  #pragma once
2
3  #include<iostream>
4  #include<string>
5
6  using namespace std;
7
8  template<class T> <T> Укажите аргументы примера шаблона для IntelliSense
9  struct Node {
10     T key;
11     Node<T>* next = nullptr;
12 };
13
14 template <class T>
15 class List {
16 private:
17     Node<T>* lastNd, * current, * head;
18     int n = 0;
19 public:
20     List() {};
21     List(int count) {
22         n = count;
23         head = new Node<T>;
24         head->key = 0;
25         lastNd = head;
26         for (int i = 1; i < n; i++) {
27             current = new Node<T>;
28             current->key = 0;
29             lastNd->next = current;
30             lastNd = current;
31         }
32         lastNd->next = NULL;
33     };
34     ~List() {
35         lastNd = head;
36         while (lastNd != NULL) {
37             current = lastNd->next;
38             delete lastNd;
39             lastNd = current;
40         }
41         n = 0;
42     };
43     List& operator = (List<T>& l) {
44         if (this != &l) {
45             if (this != 0) {
46                 lastNd = head;
47                 while (lastNd != NULL) {
48                     current = lastNd->next;
49                     delete lastNd;
50                     lastNd = current;
51                 }
52                 n = 0;
53             }

```

```

Pair.cpp  Pair.h  List.h  CL_7.cpp
CL_7  List<T>  operator=(List<T>& l)

52      n = 0;
53  }
54      lastNd = head = new Node<T>;
55      l.lastNd = l.head->next;
56      lastNd->key = l.head->key;
57      while (l.lastNd != NULL) {
58          lastNd->next = new Node<T>;
59          lastNd->next->key = l.lastNd->key;
60          l.lastNd = l.lastNd->next;
61          lastNd = lastNd->next;
62      }
63      lastNd->next = NULL;
64      n = l.n;
65  }
66      return *this;
67  };
68  int& operator[] (int index) {
69      if (index < n) {
70          lastNd = head;
71          for (int i = 0; i < index; i++) {
72              lastNd = lastNd->next;
73          }
74          return lastNd->key;
75      }
76      else {
77          cout << "Запредельный индекс";
78      }
79  };
80  int operator () () {
81      return n;
82  };
83  friend ostream& operator << (ostream& out, List<T>& l) {
84      if (l.n) {
85          l.lastNd = l.head;
86          while (l.lastNd != NULL) {
87              out << l.lastNd->key << ' ';
88              l.lastNd = l.lastNd->next;
89          }
90      }
91      else {
92          out << "Пусто";
93      }
94      return out;
95  };
96  friend istream& operator >> (istream& in, List<T>& l) {
97      l.lastNd = l.head;
98      while (l.lastNd != NULL) {
99          in >> l.lastNd->key;
100         l.lastNd = l.lastNd->next;
101     }
102     return in;
103 };

```

```
Pair.cpp  Pair.h  List.h  CL_7.cpp
CL_7      (Глобальная область)  operator>>(istream&)
```

```
103     };
104     void operator << (T t) {
105     if (head == NULL) {
106         head = new Node<T>;
107         head->key = t;
108         head->next = NULL;
109     }
110     else {
111         lastNd = head;
112         while (lastNd->next != NULL) {
113             lastNd = lastNd->next;
114         }
115         lastNd->next = new Node<T>;
116         lastNd->next->key = t;
117         lastNd = lastNd->next;
118         lastNd->next = NULL;
119     }
120     n += 1;
121     };
122 }
```

```
Pair.cpp  Pair.h  List.h  CL_7.cpp
CL_7      (Глобальная область)
```

```
1  #include <iostream>
2  #include "List.h"
3  #include "Pair.h"
4
5  using namespace std;
6
7  int main() {
8      system("chcp 1251 > null");
9
10     List<int> l;
11     l << 2;
12     l << 4;
13     cout << "List<int>:\n";
14     cout << l << '\n';
15
16     List<Pair> p;
17     Pair c;
18     c.SetFirst(16);
19     c.SetSecond(20.8);
20     p << c;
21     cout << "List<Pair>:\n";
22     cout << p << '\n';
23
24     return 0;
25 }
```

