

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 9 (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]`;

`+n` - переход вправо к элементу с номером `n`.

Код программы:

```

Error.cpp  List.cpp  Error.h  List.h  Cl_9.cpp
CL_9
List
1  #pragma once
2  #include <string>
3  #include <iostream>
4  using namespace std;
5  #include "Error.h"
6
7  struct Node {
8      int key;
9      Node* next;
10 };
11
12 class Iterator {
13 private:
14     Node* current;
15     friend class List;
16 public:
17     Iterator() {
18         current = NULL;
19     };
20     Iterator(Node* node) {
21         current = node;
22     };
23     void operator ++ () {
24         if (current == NULL) {
25             throw Error(0);
26         }
27         current = current->next;
28     }
29     Node* operator *() const {
30         return current;
31     }
32     void operator + (int shift) {
33         Node* tmp = current;
34         int i = 0;
35         while (i < shift && current != NULL && current->next != NULL) {
36             current = current->next;
37             i++;
38         }
39         if (i < shift) {
40             current = tmp;
41             throw Error(0);
42         }
43     }
44 };
45
46 class List {
47 public:
48     List();
49     List(int);
50     ~List();
51     List& operator = (List&);
52     int& operator[] (int);
53     int operator () ();
54     friend List operator*(const List&, const List&);
55     friend ostream& operator << (ostream&, List&);
56     friend istream& operator >> (istream&, List&);
57     Iterator first();
58     Iterator last();
59     void operator << (int);
60 private:
61     Node* lastNd, * current, * head;
62     int n = 0;
63     Iterator beg, end;
64 };

```

```
Error.cpp  X List.cpp  X Error.h  List.h  CL_9.cpp
CL_9
1  #include "List.h"
2
3  List::List(int count) {
4      n = count;
5      head = new Node;
6      head->key = 0;
7      lastNd = head;
8      for (int i = 1; i < n; i++) {
9          current = new Node;
10         current->key = 0;
11         lastNd->next = current;
12         lastNd = current;
13     }
14     lastNd->next = NULL;
15 }
16
17 List::~List() {
18     lastNd = head;
19     while (lastNd != NULL) {
20         current = lastNd->next;
21         delete lastNd;
22         lastNd = current;
23     }
24     n = 0;
25 }
26
27 List& List::operator=(List& l) {
28     if (this != &l) {
29         if (this != 0) {
30             lastNd = head;
31             while (lastNd != NULL) {
32                 current = lastNd->next;
33                 delete lastNd;
34                 lastNd = current;
35             }
36             n = 0;
37         }
38         lastNd = head = new Node;
39         l.lastNd = l.head->next;
40         lastNd->key = l.head->key;
41         while (l.lastNd != NULL) {
42             lastNd->next = new Node;
43             lastNd->next->key = l.lastNd->key;
44             l.lastNd = l.lastNd->next;
45             lastNd = lastNd->next;
46         }
47         lastNd->next = NULL;
48         n = l.n;
49     }
50     return *this;
51 }
```

```
Error.cpp  List.cpp  Error.h  List.h  Cl_9.cpp
Cl_9  List
operator=(List & l)

52
53 int& List::operator[](int index) {
54     if (index >= n) {
55         throw Error(3);
56     }
57     lastNd = head;
58     for (int i = 0; i < index; i++) {
59         lastNd = lastNd->next;
60     }
61     return lastNd->key;
62 }
63
64 int List::operator () () {
65     return n;
66 }
67
68 Iterator List::first() {
69     if (!n) {
70         throw Error(1);
71     }
72     beg.current = head;
73     return beg;
74 }
75
76 Iterator List::last() {
77     if (!n) {
78         throw Error(2);
79     }
80     lastNd = head;
81     while (lastNd->next != NULL) {
82         lastNd = lastNd->next;
83     }
84     end.current = lastNd;
85     return end;
86 }
87
88 ostream& operator<<(ostream& out, List& l) {
89     if (!l.n) {
90         l.lastNd = l.head;
91         while (l.lastNd != NULL) {
92             out << l.lastNd->key << ' ';
93             l.lastNd = l.lastNd->next;
94         }
95     }
96     else {
97         out << "Empty";
98     }
99     return out;
100 }
101
102 istream& operator>>(istream& in, List& l) {
103     l.lastNd = l.head;
104     while (l.lastNd != NULL) {
105         in >> l.lastNd->key;
106         l.lastNd = l.lastNd->next;
107     }
108     return in;
109 }
```

```
Error.cpp  List.cpp  Error.h  List.h  Cl_9.cpp
Cl_9
109 }
110
111 void List::operator << (int number) {
112     if (head == NULL) {
113         head = new Node;
114         head->key = number;
115         head->next = NULL;
116     }
117     else {
118         lastNd = head;
119         while (lastNd->next != NULL) {
120             lastNd = lastNd->next;
121         }
122         lastNd->next = new Node;
123         lastNd->next->key = number;
124         lastNd = lastNd->next;
125         lastNd->next = NULL;
126     }
127     n += 1;
128 }
129
130 List operator*(const List& x, const List& y) {
131     return x * y;
132 }
```

```
Error.cpp  List.cpp  Error.h  List.h  Cl_9.cpp
Cl_9 (Глобальная область)
1
2 #pragma once
3 #include <string>
4 #include <iostream>
5 using namespace std;
6
7 class Error {
8 private:
9     string code[4] = { "Error: can't shift, going off the list!", "Error: impossible get first element empty list!", "Error: impossible get last element empty list!", "Error: index outside the list!" };
10    string str;
11 public:
12     Error(int i) {
13         str = code[i];
14     }
15     void What() {
16         cout << str << '\n';
17     }
18 };
```

```
Error.cpp  List.cpp  Error.h  List.h  Cl_9.cpp
Cl_9 (Глобальная область)
1 #include <iostream>
2 #include "List.h"
3 #include "Error.h"
4
5 using namespace std;
6
7 int main() {
8     system("chcp 1251 > NULL");
9     try {
10         List a(2);
11         List b(3);
12         a[1] = 10;
13         b[2] = 20;
14         cout << a[1] * b[2] << endl;
15         cout << a << '\n';
16         cout << b << '\n';
17         cout << b[0] << endl;
18         Iterator i = b.first(); //Ошибка
19         Iterator j = b.last(); //Ошибка
20         Iterator k = a.last();
21         ++k;
22         ++k; //Ошибка
23         k + 1; //Ошибка
24     }
25     catch (Error& m_Error) {
26         m_Error.What();
27     }
28     return 0;
29 }
```