

## Лабораторная работа #1

Баженов Тимур НКАбд-02-24

```
47
48 int AddAfter(student*&university, string Newname, float Newgpa, int Newage, string NameAfter){
49     if(university==NULL){
50         cout<<"List is empty";
51         return 0;
52     }
53     student *tmp = new student;
54     student *newstudent = new student;
55     newstudent->name=Newname;
56     while(tmp!=NULL){
57         if(tmp->name==Newname){
58             cout<<"already in list";
59             return 0;
60         }
61         tmp = tmp->next;
62     }
63     tmp = university->next;
64     while(tmp!=NULL && tmp->name!=NameAfter){
65         tmp = tmp->next;
66     }
67     if(tmp==NULL){
68         return 0;
69     }
70     newstudent->age=Newage;
71     newstudent->gpa=Newgpa;
72     newstudent->next=tmp->next;
73     tmp->next=newstudent;
74     return 0;
75 }
76
77
```

Функция для добавления элемента после определенного элемента

```
int AddBefore(student*&university, string Newname, float Newgpa, int Newage, string NameBefore){
    if(university==NULL){
        return 0;
    }
    if (university->name == NameBefore){
        addfirst(university, Newname, Newgpa, Newage);
        return 0;
    }
    if(university->next == NULL){
        return 0;
    }
    student *prev = university;
    student *tmp = university->next;
    student *newstudent = new student;
    newstudent->name=Newname;

    while(tmp->next!=NULL && tmp->name!=NameBefore){
        if(tmp->name == Newname){
            cout << "already in list";
            return 0;
        }
        prev = prev->next;
        tmp = tmp->next;
    }
    if (tmp->next==NULL && tmp->name!=NameBefore){
        cout << "Element is not in list";
        return 0;
    }
    newstudent->age = Newage;
    newstudent->gpa = Newgpa;
    newstudent->next = tmp;
    prev->next=newstudent;
    return 0;
}
```

Функция для добавления элемента перед определенным элементом

```

int printName(student *university, string printName){
    bool stat = false;
    if(university==NULL){
        cout << "List is empty";
        return 0;
    }
    student *tmp = university;
    while(tmp!=NULL){
        if(tmp->name == printName){
            cout << "Student: " << tmp->name<<
                "\nGPA: " << tmp->gpa<<
                "\nAge: " << tmp->age << "\n";
            stat = true;
        }
        tmp = tmp->next;
    }
    if (stat == false){
        cout << "Element is not in list";
    }
    return 0;
}

```

Функция для удаления определенного элемента  
 Функция для вывода определенного элемента

```

int printList(student *university){
    if(university==NULL){
        cout << "List is empty";
        return 0;
    }
    student *tmp = university;
    while(tmp!=NULL){
        cout << "Student: " << tmp->name<<
            "\nGPA: " << tmp->gpa<<
            "\nAge: " << tmp->age << "\n";
        tmp = tmp->next;
    }
    return 0;
}

```

Функция для вывода всех элементов

sonal > Source Files > studentFunctions.cpp > ...

```
1  #include <iostream>
2  #include "studentFunctions.h"
3
4
5  string m_alredy_in_list = "Alredy in List";
6
7  int addfirst(student*&university,string Newname,float Newgpa, int Newage){
8      student *newstudent = new student;
9      newstudent->name = Newname;
10     student *tmp = university;
11
12     while(tmp!=NULL){
13         if(tmp->name==Newname){
14             cout<<m_alredy_in_list;
15             return 0;
16         }
17         tmp = tmp->next;
18     }
19     newstudent ->age = Newage;
20     newstudent->gpa = Newgpa;
21     newstudent->next=university;
22     university=newstudent;
23     return 0;
24 }
```

Функция для добавления элемента первым

```
5
6  int addlast(student*&university,string Newname,float Newgpa, int Newage){
7      student *tmp = university;
8      if (university==NULL){
9          addfirst(university,Newname,Newgpa,Newage);
10         return 0;
11     }
12     student *newstudent = new student;
13     newstudent->name = Newname;
14     while(tmp!=NULL){
15         if(tmp->name==Newname){
16             cout<<m_alredy_in_list;
17             return 0;
18         }
19         tmp = tmp->next;
20     }
21     newstudent ->age = Newage;
22     newstudent->gpa = Newgpa;
23     newstudent->next=NULL;
24     tmp->next=newstudent;
25     return 0;
26 }
```

Функция для добавления элемента последним

```

1  #include <iostream>
2  #include "studentFunctions.h"
3  #include <string>
4  using namespace std;
5
6
7
8
9  int main()
10 {
11     student *university = NULL;
12     int res,k,age;
13     string name,nameafter,namebefore;
14     float gpa;
15     res = 1;
16     while(res!= 0){
17         cout <<
18         "0. break\n"<<
19         "1. addfirst\n"<<
20         "2. printList\n"<<
21         "3. addLast\n"<<
22         "4. addAfter\n"<<
23         "5. addBefore\n"<<
24         "6. delElement\n"<<
25         "7. printName\n";
26         cin >> res;
27         cout << "\n";
28         switch(k){
29             case 0:{
30                 break;
31             }
32             case 1:{
33                 cin >> name>>gpa>>age;
34                 addfirst(university,name,gpa,age);
35                 break;
36             }
37             case 2:{
38                 printList(university);
39                 break;
40             }
41         }
42     }
43 }

```

Создаем бесконечный цикл

```

35     case 2:{
36         printList(university);
37         break;
38     }
39     case 3:{
40         cin >> name>>gpa>>age;
41         addlast(university,name,gpa,age);
42         break;
43     }
44     case 4:{
45         cin >> name>>gpa>>age>>nameafter;
46         AddAfter(university,name,gpa,age,nameafter);
47         break;
48     }
49     case 5:{
50         cin >> name>>gpa>>age>>namebefore;
51         AddBefore(university,name,gpa,age,namebefore);
52         break;
53     }
54
55     case 6:{
56         cin >> name;
57         delElement(university,name);
58         break;
59     }
60     case 7:{
61         cin>>name;
62         printName(university,name);
63         break;
64     }
65 }
66 }
67

```

Баженов Тимур

