Индивидуальное задание №7
Динамический список

Пример выполнения задания

Рассмотрим следующий вариант задания.

7.0. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой `Node`, содержащей два целочисленных поля `a` и `b`. Динамический однонаправленный список должен быть реализован в виде класса `List`. Поля класса: указатель на начало списка `start`, указатель на текущий элемент списка `p`. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель `p` (после добавления указатель `p` остается на месте),
2) удалить элемент, на который настроен указатель `p` (после удаления указатель `p` должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель `p`,
4) поменять значения полей данных элемента, на который настроен указатель `p`,
5) переместить указатель `p` в начало списка,
6) переместить указатель `p` на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель `p`, должен выводиться в квадратных скобках).
8) найти первый элемент списка (настроить на него указатель `p`) для которого выполняется равенство `a=b`, если таких элементов нет, то вывести сообщение «`Элементов, для которых a=b, в списке нет`».

В функции `main()` сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса `List`. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

Решение.

Рассмотрим одно из возможных решений. Сначала определим класс, описанный в задании.

```
class List {
private:
	struct Node {
		int a;
		int b;
		Node* next;
	};
	Node* start;  //указатель на первый элемент списка
	Node* p;		//указатель на текущий элемент списка
public:
	List(int x = 0, int y = 0);// конструктор класса
	~List();// деструктор класса
	void add(int x, int y);//добавить элемент
	void remove();// удалить элемент
	void show();// показать значение полей
	void change(int x, int y);//поменять значения полей
	void begin();// переместить в начало списка
	void step();//переместить указатель на один элемент вправо
	void showall();//показать весь список
};
```

Рассмотрим реализацию методов класса `List.`

1. В методе добавления элемента списка, необходимо сначала создать этот элемент и вделить под него динамическую память:

```
Node* B = new Node;
```

После этого надо заполнить поля данных нового элемента:

```
B->a = x;
B->b = y;
```

Далее настраиваем указатель в новом элементе на элемент списка, расположенный после элемента, на который настроен указатель списка:

```
B->next = p->next;
```

Настраиваем указатель предшествующего элемента на новый элемент:

```
p->next = B;
```

Полностью метод имеет вид:

```
void List::add(int x, int y) {
	Node* B = new Node;
	B->a = x;
	B->b = y;
	B->next = p->next;
	p->next = B;
};
```

2. Удаление элемента списка.

Введем новую переменную указатель на элементы списка и настроим его на начало списка:

```
Node* p1 = start;
```

Осуществляем проход по списку пока не дойдем до элемента, предшествующего тому, на который настроен указатель p:

```
while (p1->next != p) p1 = p1->next;
```

Настраиваем указатель найденного элемента на пропуск элемента, на который настроен указатель p:

```
p1->next = p->next;
```

Освобождаем память удаляемого элемента:

```
delete p;
```

Настраиваем текущий указатель списка на новый элемент:

```
p = p1;
```

Полностью метод имеет вид:

```
void List::remove() {
	Node* p1 = start;
	while (p1->next != p) p1 = p1->next;
	p1->next = p->next;
	delete p;
	p = p1;
};
```

Кроме описанных действий необходимо предусмотреть невозможность удаления последнего элемента в списке. Самостоятельно подумайте, как это сделать.

3. Вывод значений полей.

Для вывода значения полей текущего элемента списка необходимо обратиться к нему по текущему указателю списка:

```
cout << endl << "( " << p->a << ", " << p->b << "); ";
```

Полностью метод имеет вид:

```
void List::show() {
	cout << endl << "( " << p->a << ", " << p->b << "); ";
};
```

7. Изменение значения полей.

Для изменения значения полей текущего элемента осуществляем к ним доступ через указатель на текущий элемент:

```
	p->a = x;
	p->b = y;
```

Полностью метод имеет вид:

```
void List::change(int x, int y) {
	p->a = x;
	p->b = y;
};
```

5. Перемещение указателя в начало списка.

Для перемещения текущего указателя на начало списка ему необходимо присвоить значение указателя на первый элемент списка.

```
void List::begin() {
	p = start;
};
```

6. Перемещение указателя на один элемент вправо.

Перед перемещением указателя проверяем не является ли элемент последним в списке, и только после этого присвоить значение указателя, хранящегося в текущем элементе списка.

```
	if (p->next != NULL) p = p->next;
```

Полностью метод имеет вид:

```
void List::step() {
	if (p->next != NULL) p = p->next;
};
```

7. Показать весь список.

Вводим новый указатель на элементы списка и настраиваем его на начало списка:

```
	Node* p1 = start;
```

Последовательно проходим весь список и выводим его элементы, проверяя не является ли элемент текущим:

```
	cout << endl;
	while (p1 != NULL) {
		if (p1 == p) cout << "[(" << p1->a << ", " << p1->b << ")] ";
		else cout << '(' << p1->a << ", " << p1->b << "); ";
		p1 = p1->next;
```

Полностью метод имеет вид:

```
void List::showall() {
	Node* p1 = start;
	cout << endl;
	while (p1 != NULL) {
		if (p1 == p) cout << "[(" << p1->a << ", " << p1->b << ")] ";
		else cout << '(' << p1->a << ", " << p1->b << "); ";
		p1 = p1->next;
	}
};
```

8. Поиск элемента с заданными свойствами.

Метод поиска элемента реализуйте самостоятельно. Алгоритм перебора элементов списка приведен в методе `shawall()`.

```
Node* List::find(){
...
};
```

9. Конструктор класса.

Рассмотрим конструктор класса, который создает список из одного элемента с заданными значениями полей. Сначала создаем первый элемент списка:

```
	Node* B = new Node;
```

Заполняем целочисленные поля:

```
        B->a = x;
        B->b = y;
```
Помещаем в поле указателя на следующий элемент признак конца списка:
```
        B->next = NULL;
```
Настраиваем на созданный элемент указатель начала списка:
```
        start = B;
```
Настраиваем на созданный элемент текущий указатель списка:
```
        p = B;
```
Полностью метод имеет вид:
```
List::List(int x, int y) {
        Node* B = new Node;
        B->a = x;
        B->b = y;
        B->next = NULL;
        start = B;
        p = B;
};
```
10. Деструктор класса.

Рассмотрим деструктор класса, который последовательно удаляет все элементы списка. Последовательно проходим весь список, пока не дойдем до последнего элемента:
```
        p = start;
        while (p->next != NULL) p = p->next;
```
Удаляем все элементы списка, кроме первого:
```
        while (p != start) List::remove();
```
Удаляем первый элемент списка:
```
        delete start;
```
Полностью деструктор имеет вид:
```
List::~List() {
        p = start;
        while (p->next != NULL) p = p->next;
        while (p != start) List::remove();
        delete start;
}
```

Варианты заданий

7.1. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой `Node`, содержащей два целочисленных поля `a` и `b`. Динамический однонаправленный список должен быть реализован в виде класса `List`. Поля класса: указатель на начало списка `start`, указатель на текущий элемент списка `p`. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель `p` (после добавления указатель `p` остается на месте),
2) удалить элемент, на который настроен указатель `p` (после удаления указатель `p` должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель `p`,
4) поменять значения полей данных элемента, на который настроен указатель `p`,
5) переместить указатель `p` в начало списка,
6) переместить указатель `p` на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель `p2`, должен выводиться в квадратных скобках).

8) найти последний элемент списка (настроить на него указатель p) для которого выполняется равенство a=b, если таких элементов нет, то вывести сообщение «Элементов, для которых a=b, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.2. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти первый элемент списка (настроить на него указатель p) для которого выполняется неравенство a+b>c, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.3. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти последний элемент списка (настроить на него указатель p) для которого выполняется неравенство a+b>c, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.4. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:
1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти последний элемент списка (настроить на него указатель p) для которого выполняются соотношения a=0 и b≠0, если таких элементов нет, то вывести сообщение «Элементов, для которых a=0 и b≠0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.5. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:
1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти первый элемент списка (настроить на него указатель p) для которого выполняются соотношения a=0 и b≠0, если таких элементов нет, то вывести сообщение «Элементов, для которых a=0 и b≠0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.6. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти первый элемент списка (настроить на него указатель p) для которого выполняются соотношения a>0 и b≠0, если таких элементов нет, то вывести сообщение «Элементов, для которых a>0 и b≠0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.7. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти последний элемент списка (настроить на него указатель p) для которого выполняются соотношения a=0 и b≠0, если таких элементов нет, то вывести сообщение «Элементов, для которых a=0 и b≠0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.8. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти последний элемент списка (настроить на него указатель p) для которого выполняются соотношения a>0 и b≠0, если таких элементов нет, то вывести сообщение «Элементов, для которых a>0 и b≠0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.9. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти первый элемент списка (настроить на него указатель p) для которого выполняются соотношения a>0 и b>a, если таких элементов нет, то вывести сообщение «Элементов, для которых a>0 и b>a, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.10. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти последний элемент списка (настроить на него указатель p) для которого выполняются соотношения a>0 и b>a, если таких элементов нет, то вывести сообщение «Элементов, для которых a>0 и b>a, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.11. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти последний элемент списка (настроить на него указатель p) для которого выполняется равенство a*b=0, если таких элементов нет, то вывести сообщение «Элементов, для которых a*b=0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.12. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти первый элемент списка (настроить на него указатель p) для которого выполняется равенство a*b=0, если таких элементов нет, то вывести сообщение «Элементов, для которых a*b=0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.13. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти первый элемент списка (настроить на него указатель p) для которого выполняется неравенство a*b>0, если таких элементов нет, то вывести сообщение «Элементов, для которых a*b>0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.14. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти последний элемент списка (настроить на него указатель p) для которого выполняется неравенство a*b>0, если таких элементов нет, то вывести сообщение «Элементов, для которых a*b>0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.15. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти первый элемент списка (настроить на него указатель p) для которого выполняется неравенство a+b>0, если таких элементов нет, то вывести сообщение «Элементов, для которых a+b>0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.16. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти последний элемент списка (настроить на него указатель p) для которого выполняется неравенство a+b>0, если таких элементов нет, то вывести сообщение «Элементов, для которых a+b>0, в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.17. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти первый элемент списка (настроить на него указатель p) для которого выполняется неравенство a*b>c, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.18. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти первый элемент списка (настроить на него указатель p) для которого выполняется неравенство a*b<c, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.19. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти первый элемент списка (настроить на него указатель p) для которого выполняется неравенства a>c и b<c, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.20. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти первый элемент списка (настроить на него указатель p) для которого выполняется неравенства a>c или b<c, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.21. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти последний элемент списка (настроить на него указатель p) для которого выполняется неравенства a>c и b<c, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.22. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти последний элемент списка (настроить на него указатель p) для которого выполняется неравенства a>c или b<c, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.23. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти последний элемент списка (настроить на него указатель p) для которого выполняется неравенства a>c и b<0, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.24. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).
8) найти первый элемент списка (настроить на него указатель p) для которого выполняется неравенства a>c и b<0, где c – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «Таких элементов в списке нет».

В функции main() сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса List. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».

7.25. Напишите программу, работающую с динамическим однонаправленным списком. Элемент списка должен описываться структурой Node, содержащей два целочисленных поля a и b. Динамический однонаправленный список должен быть реализован в виде класса List. Поля класса: указатель на начало списка start, указатель на текущий элемент списка p. Конструктор класса должен создавать список из одного элемента, деструктор класса должен уничтожать список. Кроме этого в классе должны присутствовать методы:

1) добавить элемент после элемента, на который настроен указатель p (после добавления указатель p остается на месте),
2) удалить элемент, на который настроен указатель p (после удаления указатель p должен быть настроен на элемент, предшествующий удаляемому),
3) показать значения полей данных элемента, на который настроен указатель p,
4) поменять значения полей данных элемента, на который настроен указатель p,
5) переместить указатель p в начало списка,
6) переместить указатель p на один элемент вправо,
7) вывести на консоль значение всех элементов списка (элемент, на который настроен указатель p, должен выводиться в квадратных скобках).

8) найти последний элемент списка (настроить на него указатель `p`) для которого выполняется неравенства `a*b>c` и `b<0`, где `c` – число, запрашиваемое у пользователя. Если таких элементов нет, то вывести сообщение «`Таких элементов в списке нет`».

В функции `main()` сначала запрашиваются значения полей первого элемента списка, после чего создается объект класса `List`. После этого выводится меню, позволяющее вызывать методы объекта любое количество раз, пока пользователь не выберет пункт меню «Выход из программы».