

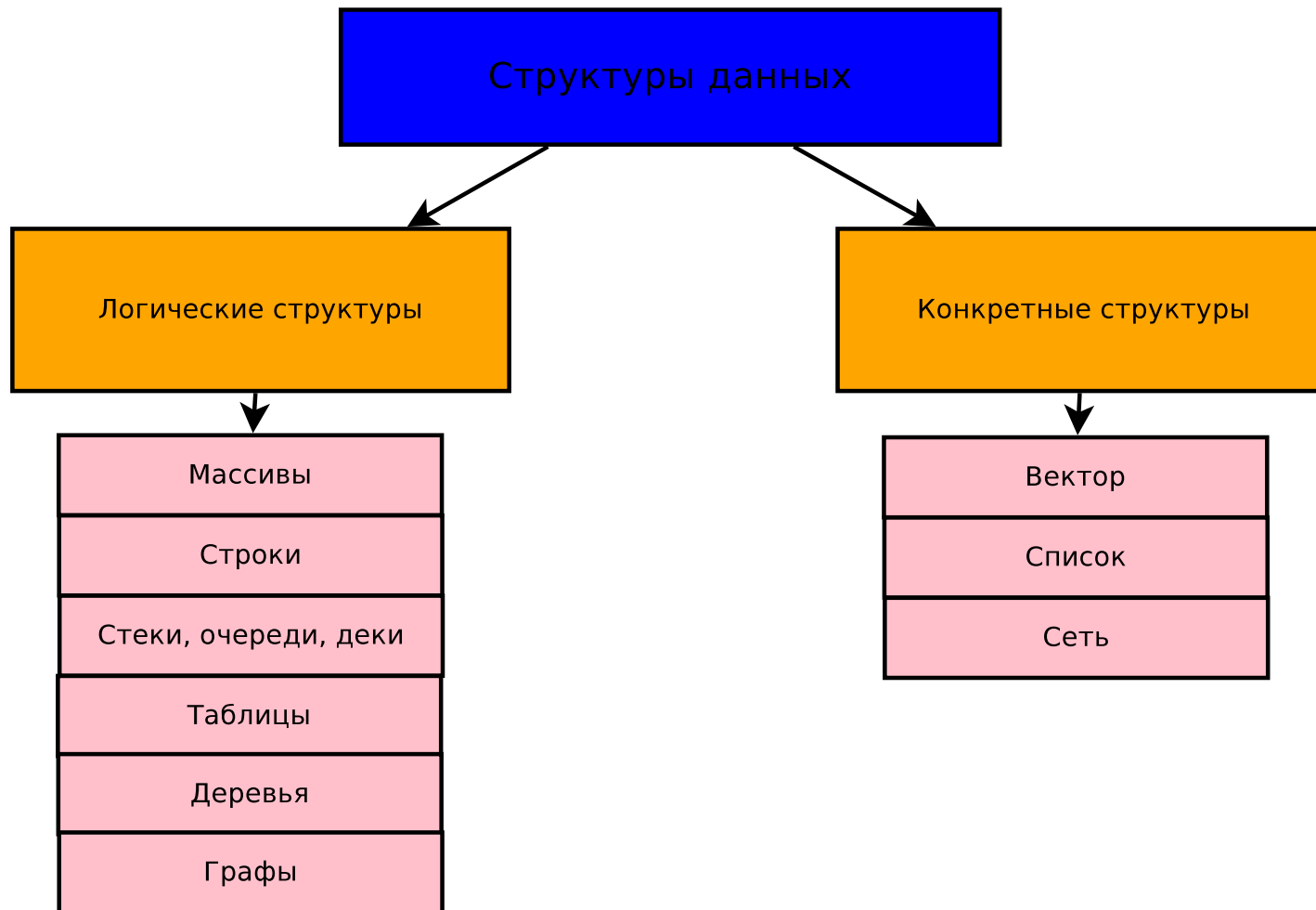
# Структуры данных

Артамонов Ю.Н.

Международный университет  
природы, общества и человека "Дубна"  
филиал Котельники

5 апреля 2016 г.

# Логические и конкретные структуры



Логическая структура - это множество элементов и связей между ними  
Конкретная структура - это способ представления логической структуры в памяти ЭВМ

# Операции с логическими структурами

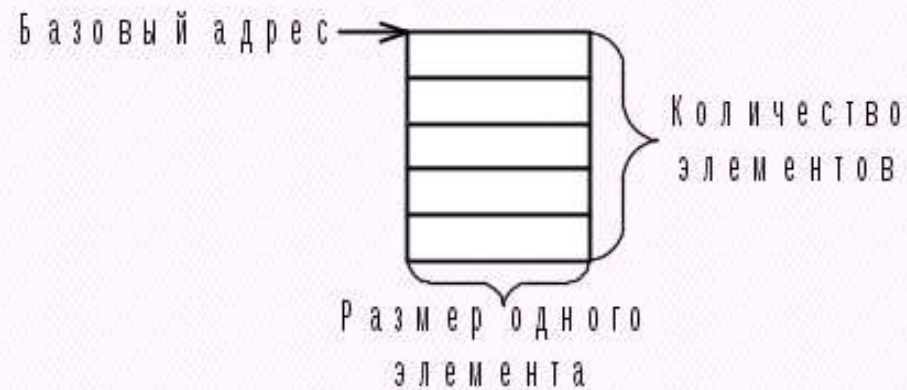
Запрос	Модификация
Search(S,k)	Insert(S,x)
Minimum(S)	Delete(S,x)
Maximum(S)	—
Successor(S,x)	—
Predecessor(S,x)	—

S - множество объектов

x - указатель на элемент множества

k - значение ключа

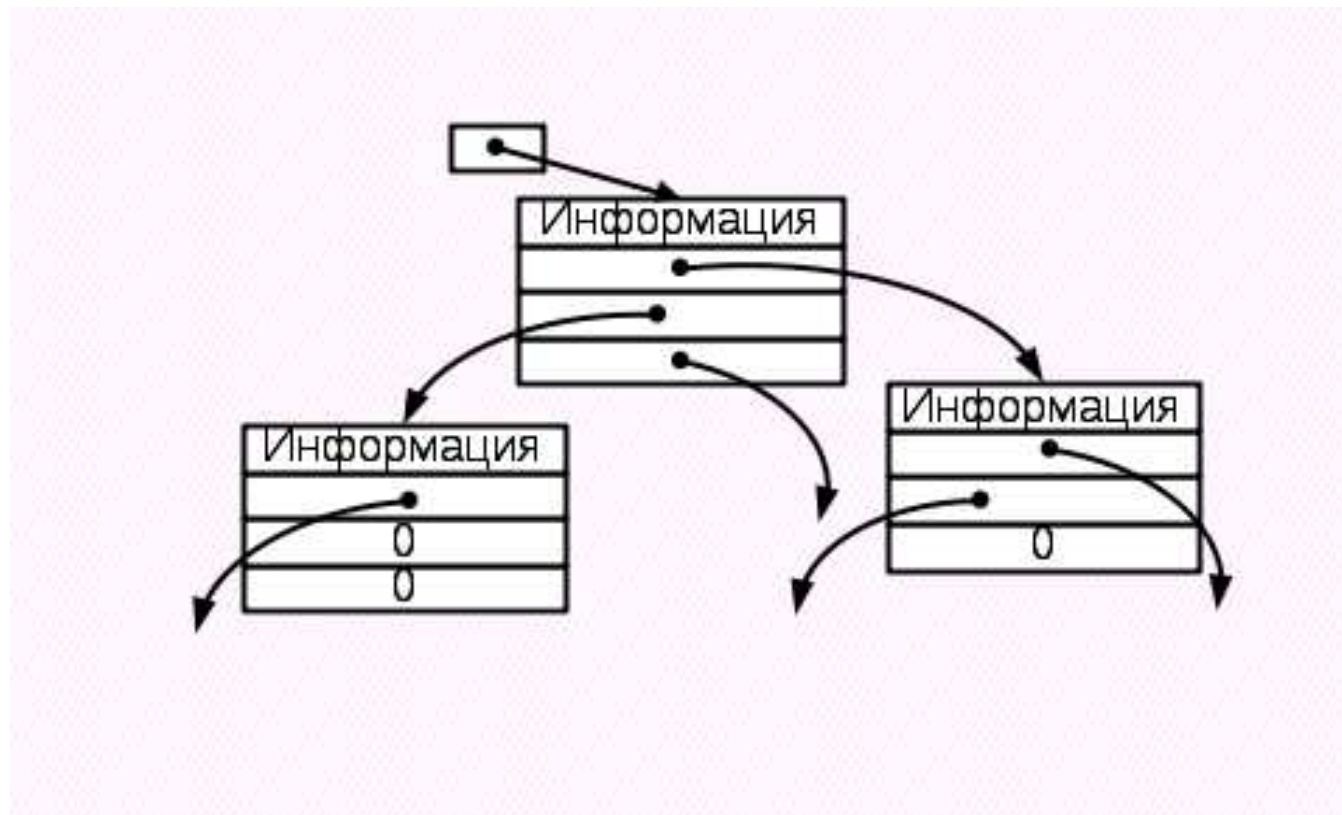
# Вектор



Доступ к элементу – по индексу

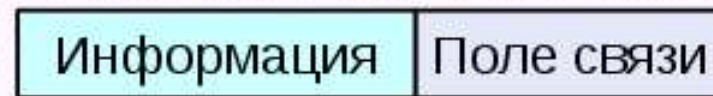
$$\text{Адрес}_i = \text{базовый\_адрес} + i * \text{размер\_элемента}$$

# Сеть



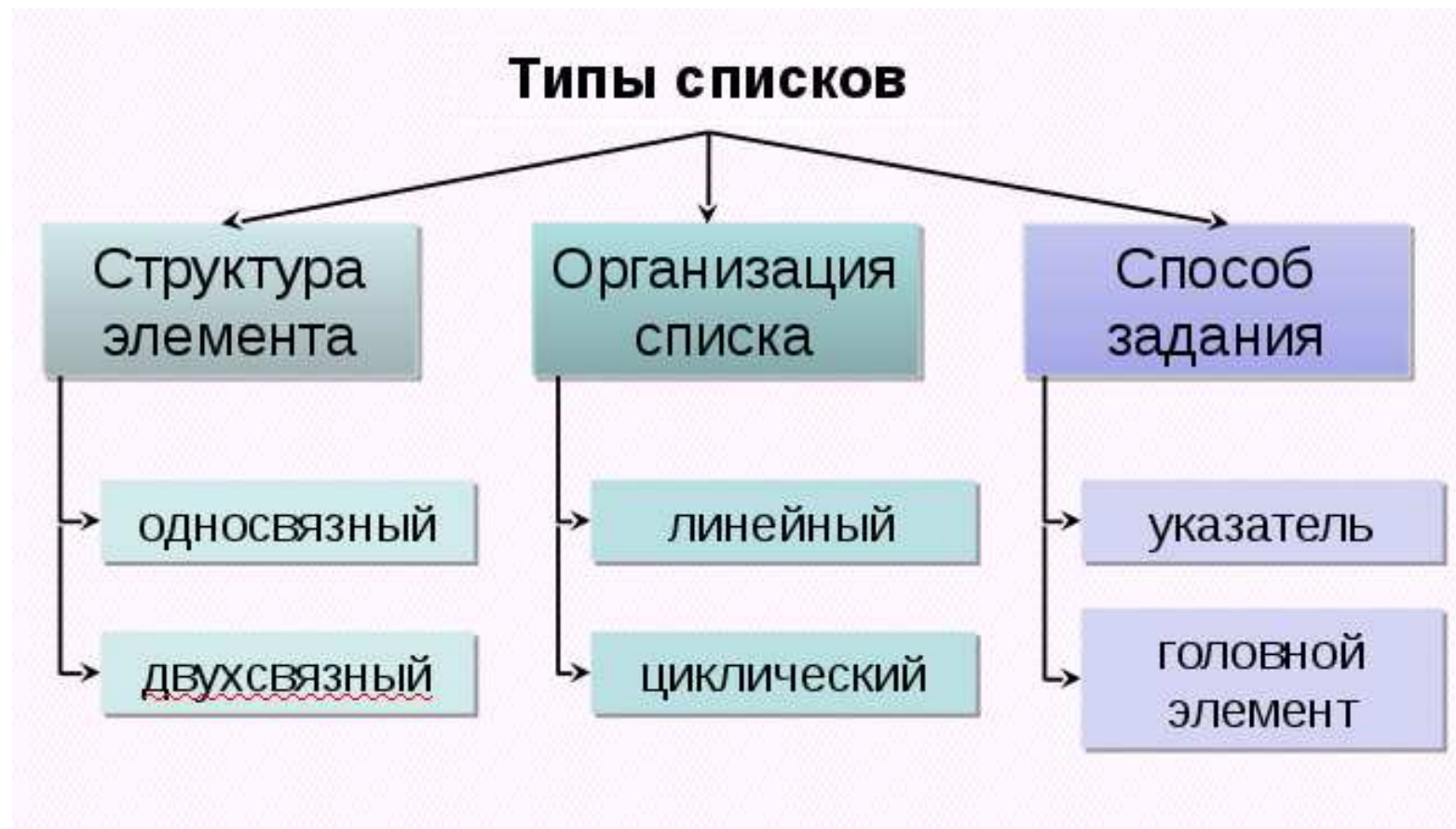
# Элемент списка Node

Элемент списка



Доступ к элементу – по указателю

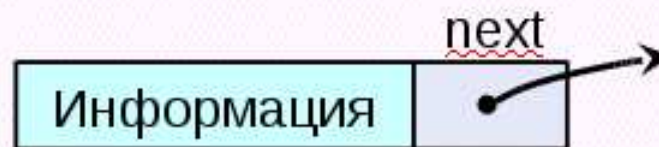
# Типы списков



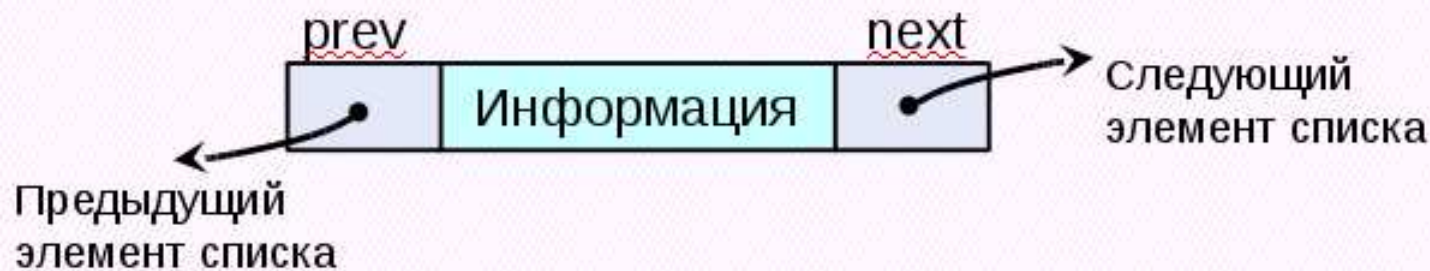


# Структура элемента списка

- Односвязный список



- Двухсвязный список



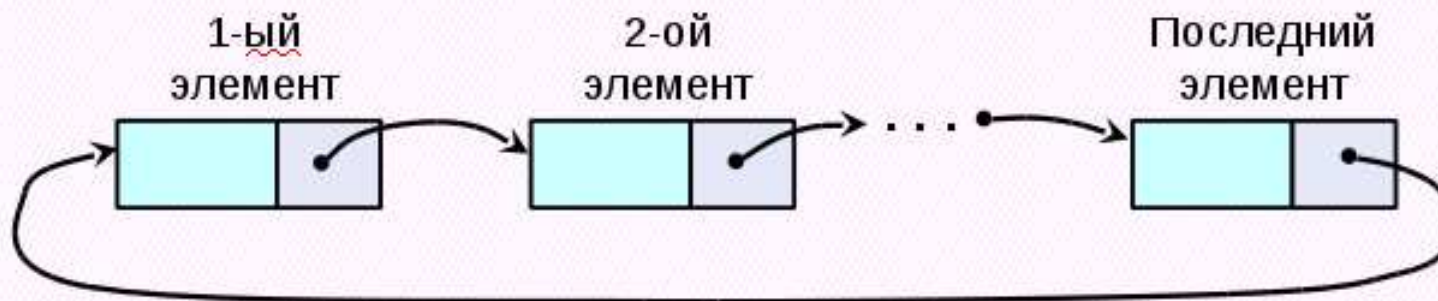


# Способ организации списка

- **Линейный список**



- **Циклический список**

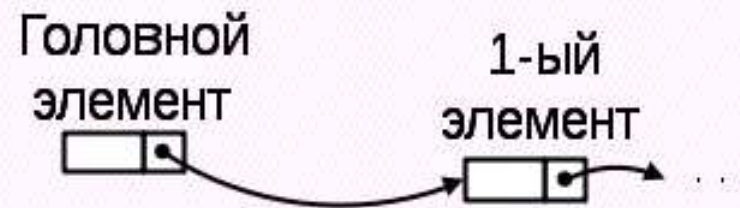


# Способ задания списка

- Указатель на начало списка



- Головной элемент

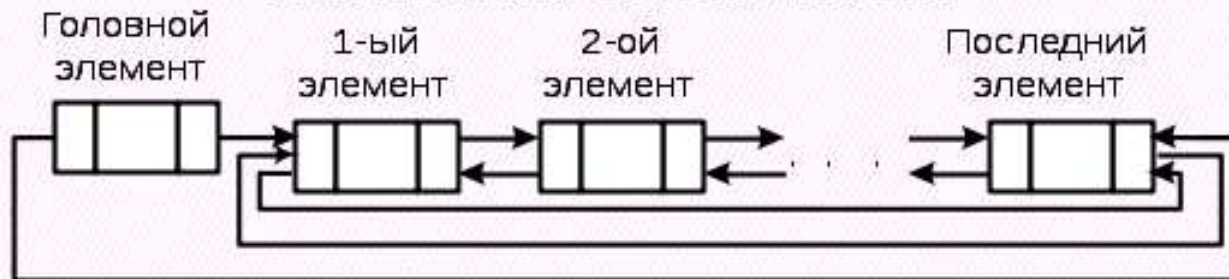


# Примеры списков

## Линейный односвязный список с головным элементом



## Циклический двухсвязный список с головным элементом



# Реализация односвязного списка в Python

```
class Node:  
    __value = None  
    __next = None  
  
    def set_value(self, x):  
        return self.__value = x  
  
    def set_next(self, y):  
        return self.__next = y  
  
    def get_value(self):  
        return self.__value  
  
    def get_next(self):  
        return self.__next
```

# Реализация односвязного списка в Python (продолжение)

```
class mylist:
    __first = None
    __count = 0

    def add(self, x):
        a = Node()
        a.set_value(x)
        a.set_next(self.__first)
        self.__first = a
        self.__count += 1
        del a

    def get_count(self):
        return self.__count
```

# Реализация односвязного списка в Python (продолжение)

```
def get_first(self):  
    return self.__first  
  
def print_list(self):  
    x = self.__first  
    while x.get_next() != None:  
        print(x.get_value())  
        x = x.get_next()  
    print(x.get_value())
```

# Пример работы со списком

```
L = mylist()
```

```
L.add(0)
```

```
L.add(7)
```

```
L.add(4)
```

```
L.print_list()
```

```
4
```

```
7
```

```
0
```

```
L.get_count()
```

```
3
```



# Задания

1. В списке реализовать метод удаления элемента с головы списка
2. В списке реализовать метод удаления элемента с конца списка
3. В списке реализовать вставку элемента в заданную позицию
4. В списке реализовать поиск заданного элемента
5. В списке реализовать пузырьковую сортировку