

## **Лабораторная работа № 6**

### **Шаблоны функций и классов**

#### **Вариант 1**

Разработать класс List (однонаправленный список). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в начало списка (item + list);
- 2) удаление элемента из начала списка (--list);
- 3) проверка на равенство списков (==).

#### **Вариант 2**

Разработать класс Stack (стек). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в стек (+);
- 2) извлечение элемента из стека (--);
- 3) сравнение стеков по количеству элементов (<=).

#### **Вариант 3**

Разработать класс Queue (очередь). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) объединение очередей (+);
- 2) извлечение элемента из очереди (--);
- 3) сравнение очередей по количеству элементов (>=).

#### **Вариант 4**

Разработать класс List (однонаправленный список). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в конец списка (list + item);
- 2) удаление элемента из конца списка (list--);
- 3) проверка на равенство списков (==).

#### **Вариант 5**

Разработать класс Stack (стек). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) объединение стеков (+);
- 2) извлечение элемента из стека (--);
- 3) сравнение стеков по количеству элементов (<=).

### **Вариант 6**

Разработать класс Queue (очередь). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в очередь (+);
- 2) извлечение элемента из очереди (--);
- 3) проверка на равенство очередей (!=).

### **Вариант 7**

Разработать класс List (однонаправленный список). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в заданную позицию (например: int i; Type c; list L; ...; L(c,i); );
- 2) удаление элемента из конца списка (list--).

### **Вариант 8**

Разработать класс Stack (стек). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в стек (+);
- 2) извлечение элемента из стека (--);
- 3) сравнение стеков по количеству элементов (>).

### **Вариант 9**

Разработать класс Queue (очередь). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в очередь (+);
- 2) извлечение элемента из очереди (--);
- 3) сравнение очередей по количеству элементов (<).

### **Вариант 10**

Разработать класс List (однонаправленный список). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в начало списка (item + list);
- 2) удаление элемента из заданной позиции (например: int i; list L; ...; L(i); ).

### **Вариант 11**

Разработать класс Stack (стек). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в стек (+);
- 2) извлечение элемента из стека (--);
- 3) сравнение стеков по количеству элементов (<).

### **Вариант 12**

Разработать класс Queue (очередь). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в очередь (+);
- 2) извлечение элемента из очереди (--);
- 3) сравнение очередей по количеству элементов (>).

### **Вариант 13**

Разработать класс List (однонаправленный список). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в заданную позицию (например: int i; Type c; list L; ...; L(c,i); );
- 2) удаление элемента из начала списка (--list).

### **Вариант 14**

Разработать класс Stack (стек). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) объединение стеков (+);
- 2) извлечение элемента из стека (--);
- 3) проверка на равенство стеков (!=).

### **Вариант 15**

Разработать класс Queue (очередь). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в очередь (+);
- 2) извлечение элемента из очереди (--);
- 3) сравнение очередей по количеству элементов (<=).

### **Вариант 16**

Разработать класс List (однонаправленный список). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в конец списка (list + item);
- 2) удаление элемента из заданной позиции (например: int i; list L; ...; L(i); ).

### **Вариант 17**

Разработать класс Stack (стек). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в стек (+);
- 2) извлечение элемента из стека (--);
- 3) проверка на равенство стеков (!=).

### **Вариант 18**

Разработать класс Queue (очередь). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в очередь (+);
- 2) извлечение элемента из очереди (--);
- 3) сравнение очередей по количеству элементов (>=).

### **Вариант 19**

Разработать класс List (однонаправленный список). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в заданную позицию (например: int i; Type c; list L; ...; L(c,i); );
- 2) удаление элемента из заданной позиции (например: int i; list L; ...; L(i); ).

### **Вариант 20**

Разработать класс Stack (стек). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для

пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) объединение стеков (+);
- 2) извлечение элемента из стека (--);
- 3) проверка на равенство стеков (==).

### **Вариант 21**

Разработать класс Queue (очередь). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в очередь (+);
- 2) извлечение элемента из очереди (--);
- 3) проверка на равенство очередей (==).

### **Вариант 22**

Разработать класс List (однонаправленный список). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в конец списка (list + item);
- 2) удаление элемента из конца списка (list--);
- 3) проверка на неравенство списков (!=).

### **Вариант 23**

Разработать класс Stack (стек). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в стек (+);
- 2) извлечение элемента из стека (--);
- 3) сравнение стеков по количеству элементов (>=).

### **Вариант 24**

Разработать класс Queue (очередь). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) объединение очередей (+);
- 2) извлечение элемента из очереди (--);
- 3) проверка на равенство очередей (==).

### **Вариант 25**

Разработать класс List (однонаправленный список). В качестве стандартных типов использовать целочисленный, вещественный и

символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в начало списка (list + item);
- 2) удаление элемента из начала списка (--list);
- 3) проверка на неравенство списков (!=).

### **Вариант 26**

Разработать класс Stack (стек). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) добавление элемента в стек (+);
- 2) извлечение элемента из стека (--);
- 3) проверка на равенство стеков (==).

### **Вариант 27**

Разработать класс Queue (очередь). В качестве стандартных типов использовать целочисленный, вещественный и символьный типы. Для пользовательского типа использовать класс из лабораторной работы № 3. Перегрузить следующие операции:

- 1) объединение очередей (+);
- 2) извлечение элемента из очереди (--);
- 3) проверка на неравенство очередей (==).