Задания

Общие указания для всех вариантов

Во всех вариантах требуется создать шаблон некоторого целевого класса A, возможно, реализованный с применением некоторого серверного класса B. Это означает, что объект класса B используется как элемент класса A. В качестве серверного класса может быть указан либо класс, созданный программистом[1], либо класс из стандартной библиотеки — например, `std::vector`.

Таблица 3.1. Варианты целевых или серверных классов

Имя класса	Назначение
`Vect`	одномерный динамический массив
`List`	двунаправленный список

[1] В рамках этого же задания.

Имя класса	Назначение
Stack	стек
BinaryTree	бинарное дерево
Queue	односторонняя очередь
Deque	двусторонняя очередь (допускает вставку и удаление из обоих концов очереди)
Set	множество (повторяющиеся элементы в множество не заносятся; элементы в множестве хранятся отсортированными)
SparseArray	разреженный массив

Если вместо серверного класса указан динамический массив, то это означает, что для хранения элементов контейнерного класса используется массив, размещаемый с помощью операции new.

Во всех вариантах необходимо предусмотреть генерацию и обработку исключений для возможных ошибочных ситуаций.

Во всех вариантах показать в клиенте main() использование созданного класса, включая ситуации, приводящие к генерации исключений. Показать инстанцирование шаблона для типов int, double, std::string.

Варианты заданий приведены в табл. 3.2.

Таблица 3.2. Варианты заданий

Вариант	Целевой шаблонный класс	Реализация с применением
1	Vect	std::list
2	List	—
3	Stack	динамический массив
4	Stack	Vect
5	Stack	List
6	Stack	std::vector
7	Stack	std::list
8	BinaryTree	—
9	Queue	Vect
10	Queue	List
11	Queue	std::list
12	Deque	Vect
13	Deque	List
14	Deque	std::list
15	Set	динамический массив
16	Set	Vect
17	Set	List
18	Set	std::vector
19	Set	std::list
20	SparseArray	List