**Варианты индивидуальных заданий**

**на лабораторную работу №6**

**(изучение шаблонных классов)**

**Вариант 1**

Разработать шаблонный класс CArray, который предназначен для реализации безопасного одномерного массива фиксированной длинны. Размер массива, как и тип его элементов, должен быть параметром шаблона. Обязательно наличие конструктора по умолчанию.

Контроль границ массива должен быть организован с помощью перегрузки оператора индексации. Предусмотреть операторные функции для перегрузки операторов ввода и вывода.

**Вариант 2**

Разработать шаблонный класс CPair с двумя параметрами типа, которые определяют два различных типа двух полей этого шаблонного класса.

Обязательно наличие следующих методов:

* конструктор с параметрами;
* конструктор по умолчанию;
* методы для задания и получения значений первого и второго поля.

Предусмотреть операторные функции для перегрузки операторов ввода и вывода.

**Вариант 3**

Разработать шаблонный класс CArray, предназначенный для реализации безопасного одномерного динамического массива. Для хранения элементов массива должна быть использована одноуровневая непрямая адресация с динамическим выделением памяти под массив в конструкторе с параметрами (в качестве параметров – указатель на исходный массив и количество элементов массива). Для хранения количества элементов массива объявить целочисленное поле. Обязательно наличие следующих методов:

* конструктор с параметрами;
* конструктор копирования;
* конструктор по умолчанию;
* деструктор.

Предусмотреть операторные функции для перегрузки операторов присваивания, ввода и вывода.

**Вариант 4**

Разработать шаблонный класс CMatrix для реализации матрицы m х n. Количество строк и столбцов матрицы, как и тип его элементов, должны быть параметрами шаблона. Обязательно наличие следующих методов:

* конструктор по умолчанию;
* метод для присвоения элементу матрицы с определенным номером строки и столбца некоторого значения;
* метод получения значения элемента матрицы по номеру строки и столбца.

Предусмотреть операторные функции для перегрузки операторов ввода и вывода.

**Вариант 5**

Разработать шаблонный класс CMatrix, предназначенный для реализации матрицы m x n. Для хранения элементов массива должна быть использована двухуровневая непрямая адресация с динамическим выделением памяти под матрицу в конструкторе с параметрами (в качестве параметров – количество строк и столбцов матрицы). Для хранения количества строк и столбцов матрицы объявить два целочисленных поля. Обязательно наличие следующих методов:

* конструктор с параметрами;
* конструктор копирования;
* деструктор;
* метод для присвоения элементу матрицы с определенным номером строки и столбца некоторого значения;
* метод получения значения элемента матрицы по номеру строки и столбца.

Предусмотреть операторные функции для перегрузки операторов присваивания, ввода и вывода.

**Вариант 6**

Разработать шаблонный класс CStack для реализации стека. Для хранения элементов стека должна быть использован одномерный статический массив. Объявить целочисленное поле – для хранения индекса вершины стека. Количество элементов стека, как и их тип, должны быть параметрами шаблона. Обязательно наличие следующих методов:

* конструктор по умолчанию;
* метод для добавления элемента в стек;
* метод извлечения элемента из стека;
* методы проверки полноты и пустоты стека.

**Вариант 7**

Разработать шаблонный класс CQueue для реализации очереди. Для хранения элементов очереди должна быть использована одноуровневая непрямая адресация с динамическим выделением памяти под массив в конструкторе с параметрами (в качестве целочисленного параметра – количество элементов очереди). Объявить три целочисленных поля – для хранения текущего количества элементов очереди и для индексов хвоста и головы очереди. Обязательно наличие следующих методов:

* конструктор с параметрами
* конструктор с копирования;
* деструктор;
* метод для добавления элемента в очередь;
* метод извлечения элемента из очереди;
* методы проверки полноты и пустоты очереди.

Предусмотреть операторную функцию для перегрузки оператора присваивания.

**Вариант 8**

Разработать шаблонный класс CStack для реализации стека. Для хранения элементов стека должна быть использована одноуровневая непрямая адресация с динамическим выделением памяти под массив в конструкторе с параметрами (в качестве целочисленного параметра – количество элементов стека). Объявить два целочисленных поля – для хранения текущего количества элементов стека и для индекса вершины стека. Обязательно наличие следующих методов:

* конструктор с параметрами;
* конструктор копирования;
* деструктор;
* метод для добавления элемента в стек;
* метод извлечения элемента из стека;
* методы проверки полноты и пустоты стека.

Предусмотреть операторную функцию для перегрузки оператора присваивания.

**Вариант 9**

Разработать шаблонный класс CQueue для реализации очереди. Для хранения элементов должен быть использован одномерный статический массив. Количество элементов очереди, как и их тип, должны быть параметрами шаблона.

Обязательно наличие следующих методов:

* конструктор с параметрами;
* деструктор;
* метод для добавления элемента в очередь;
* метод извлечения элемента из очереди.

Предусмотреть операторную функцию для перегрузки оператора сравнения.

**Вариант 10**

Разработать шаблонный класс CDataSize с одним параметром типа, который определяет тип единственного поля этого шаблонного класса.

Обязательно наличие следующих методов:

* конструктор с параметрами;
* методы для задания и получения значений поля;
* метод, который возвращает размер памяти поля шаблонного класса CDataSize.

Предусмотреть операторные функции для перегрузки операторов ввода и вывода.