Практическая работа. Абстрактный тип данных (ADT) Множество (на шаблоне)

Тема: Классы C++, библиотека STL

Цель: Сформировать практические навыки: реализации абстрактных типов данных с помощью классов C++, шаблонов и библиотеки шаблонов STL, ассоциативного контейнера set

Задание

1. В соответствии с приведенной ниже спецификацией реализуйте шаблон классов «множество». Для тестирования в качестве параметра шаблона T выберите типы:
   - int;
   - TFrac (простая дробь), разработанный вами ранее.
2. Протестируйте каждую операцию, определенную на типе данных одним из методов тестирования.

Спецификация типа данных «множество»

ADT tset

Данные

Множества - это изменяемые неограниченные наборы элементов типа T. Содержимое множества изменяется следующими операциями:

- Опустошить (опустошение множества);
- Добавить (добавление элемента во множество);
- Удалить (извлечение элемента из множества).

Множество поддерживает следующую дисциплину записи и извлечения элементов: элемент может присутствовать во множестве только в одном экземпляре, при извлечении выбирается заданный элемент множества и удаляется из множества.

Операции

Операции могут вызываться только объектом «множество» (тип tset), указатель на который передаётся в них по умолчанию. При описании операций этот объект в разделе «Вход» не указывается.

Таблица 1. Описание операций на ADT tset.

Наименование Операции	Описание
Конструктор	
Начальные	Нет.

значения:	
Процесс:	Создаёт пустое множество элементов типа T.
Опустошить	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Удаляет из множества все элементы.
Выход:	Нет.
Постусловия:	Множество - пусто.
Добавить	
Вход:	d – элемент типа T.
Предусловия:	Нет.
Процесс:	Добавляет d во множество, если в нем нет такого элемента.
Выход:	Нет.
Постусловия:	Множество содержит элемент d.
Удалить	
Вход:	d – элемент типа T.
Предусловия:	Нет.
Процесс:	Удаляет элемент d из множества, если d принадлежит множеству.
Выход:	Нет.
Постусловия:	Множество не содержит элемент d.
Пусто	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Определяет, содержит ли множество элементы. Возвращает значение True, если множество не пусто, False – в противном случае.
Выход:	Булевское значение.
Постусловия:	Нет.
Принадлежит	
Вход:	d – элемент типа T.
Предусловия:	Нет.
Процесс:	Определяет, принадлежит ли элемент d множеству. Возвращает True, если d принадлежит множеству, False - в противном случае.

Выход:	Булевское значение.
Постусловия:	Нет.
Объединить	
Вход:	Множество q.
Предусловия:	Нет
Процесс:	Создаёт множество, полученное в результате объединения множества с множеством q.
Выход:	Множество.
Постусловия:	Нет.
Вычесть	
Вход:	Множество q.
Предусловия:	Нет.
Процесс:	Создаёт множество, полученное в результате вычитания из множества множество q.
Выход:	Множество.
Постусловия:	Нет.
Умножить	
Вход:	Множество q.
Предусловия:	Нет.
Процесс:	Создаёт множество, являющееся пересечением множества с множеством q.
Выход:	Множество.
Постусловия:	Нет.
Элементов	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Подсчитывает и возвращает количество элементов во множестве, если множество пустое - ноль
Выход:	Целое - количество элементов во множестве.
Постусловия:	Нет.
Элемент	
Вход:	j - номер элемента множества.
Предусловия:	Нет.
Процесс:	Обеспечивает доступ к элементу

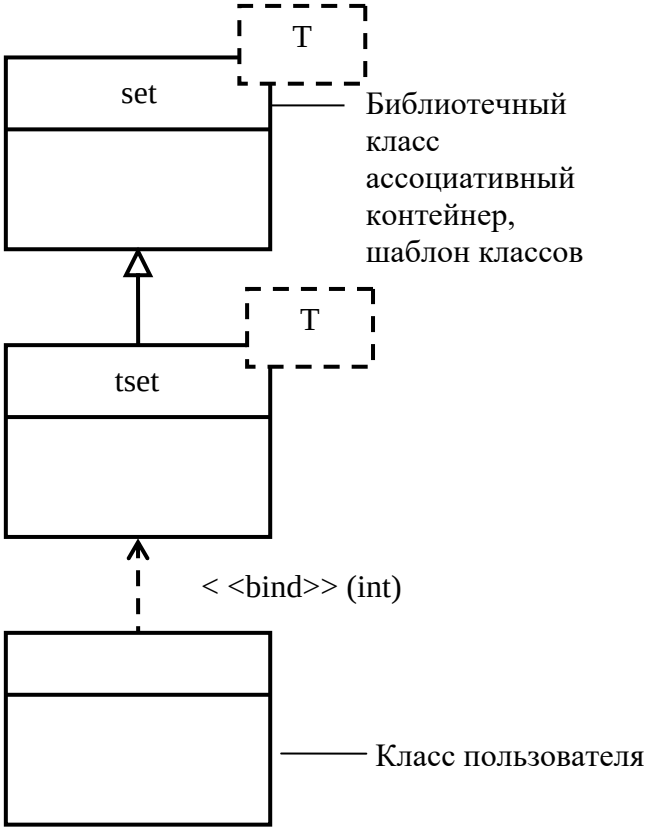
	множества для чтения по индексу j так, что если изменять j от 1 до количества элементов во множестве, то можно просмотреть все элементы множества.
Выход:	Элемент множества типа T.
Постусловия:	Множество не модифицируется

end tset

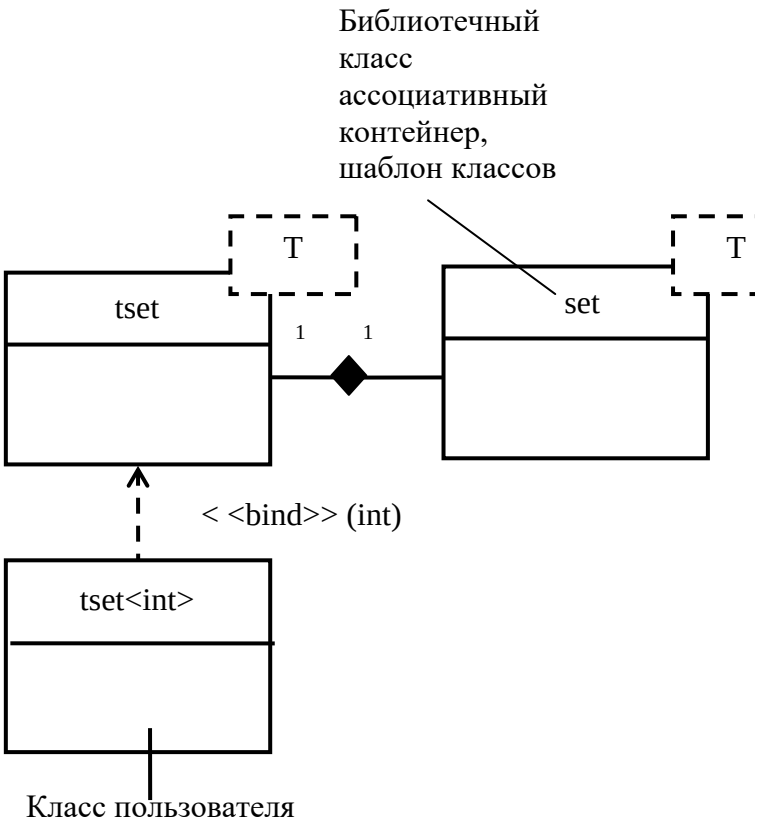
Рекомендации к выполнению

1. Тип данных реализуйте, используя ассоциативный контейнер множество (set).
2. Шаблон классов «множество» реализуйте двумя способами:

наследуя от ассоциативного контейнера set, как представлено ниже:



описав поле типа ассоциативный контейнер set<T> класса «множество», как показано на рисунке ниже:



Для обработки объектов с помощью множества tset, пользователю необходимо будет вместо идентификатора T подставить тип его объектов.

3. Тип данных реализуйте в отдельном файле utset.

Порядок выполнения

Реализуйте заданную абстракцию данных по обоим вариантам в режиме консольного приложения:

1. Создайте консольное приложение и сохраните его под именем pset.
2. Добавьте к консольному приложению файл и сохраните его под именем uset.
3. В модуле uset опишите шаблон классов tset.
4. Подключите файл к консольному приложению с помощью директивы #include.
5. Разработайте тестовый набор данных для тестирования операций, заданных на множестве. Тестовый набор поместите в таблицу следующего вида:

Таблица 2. Тестовый набор для тестирования шаблона классов «множество».

Тестовый набор для тестирования операции Сложить множества целых чисел				
Номер теста	Исходные данные		Ожидаемый результат	
	Вход	Множество	Возвращаемое значение	Множество
1	()	()	()	()
2	(0)	()	(0)	()
3	(1)	(0)	(1 0)	(0)
4	(1 0)	(1 0)	(1 0)	(1 0)
5	(1 2 3)	(3 4 5)	(1 2 3 4 5)	(3 4 5)

6. Протестируйте разработанную абстракцию данных в режиме консольного приложения.

Контрольные вопросы

1. В каком файле описан ассоциативный контейнер set?
2. Что означает имя iterator в области видимости ассоциативного контейнера set?
3. Назначение метода insert() ассоциативного контейнера set?
4. Назначение метода erase() ассоциативного контейнера set?
5. Назначение метода empty() ассоциативного контейнера set?
6. Назначение метода clear() ассоциативного контейнера set?
7. Назначение метода size() ассоциативного контейнера set?
8. Назначение метода find() ассоциативного контейнера set?
9. В чём особенности абстрактных методов?
10. В чём особенности ассоциативный контейнер в отличие от последовательных контейнеров?
11. Как пользоваться реализованными вами шаблонами классов «множество»?