

Лабораторная работа по классу-контейнеру Set

Задание на STL класс-контейнер Set

Необходимо задать три множества: универсальное множество (U) и два множества A и B, входящие в него. Затем произвести множественные операции над множествами, получить новое множество, и вывести часть элементов полученного множества, удовлетворяющих определенному условию на экран

Параметры задания:

1. Способ задания универсального множества
2. Способ выбора элементов множества A из универсального множества
3. Способ выбора элементов множества B из универсального множества
4. Операции над множествами A и B (отрицание, пересечение, объединение, разность, симметрическая разность)
5. Признак элементов, которые будут выведены на экран (все элементы, первый, последний, из диапазона)

Подробное описание параметров задания:

1. Способ задания универсального множества U:

1. Ввод с консоли
2. Множество всех натуральных чисел из диапазона [a, b]
3. N случайных чисел из диапазона [a, b];
4. Формула $\{x: t \in [a, b], x = K * t^n\}$ ($n = 1$ или 2)
5. Числа Фибоначчи до N-го члена

2. Способ выбора элементов множества A из универсального множества

1. Выбрать вручную
2. Случайным образом (случайным образом выбираются индексы универсального множества)
3. Четные по значению элементы универсального множества
4. Нечетные по значению элементы
5. Элементы, принадлежащие диапазону [c, d]
6. Элементы, которые кратны K_1 или K_2

Примечание: случайные числа генерируются с помощью функции rand(). Генерация чисел из диапазона [a,b] производится следующим образом: $R = \text{rand()} \% (b - a + 1) + a$. Для того, чтобы случайны числа не повторялись каждый раз при новом запуске программы необходимо включить в программу заголовочный файл

#include <time.h>

и поместить в начале программы команду:

srand(time(NULL));

3. Способы выбора элементов множества В из универсального множества

1. Выбрать вручную
2. Случайным образом (случайным образом выбираются индексы универсального множества)
3. Четные по значению элементы
4. Нечетные по значению элементы
5. Элементы, принадлежащие диапазону [c, d]
6. Элементы, которые кратны K_1 или K_2

Примечание: следует выбирать подмножества А и В таким образом, чтобы $A \cap B \neq \emptyset$ и $A \neq B$

4. Реализовать два преобразования над множествами А и В и показать идентичность их результата

1. $A \cap B$ и $\neg(A \cup B)$
2. $\neg(A \cap B)$ и $A \cup B$
3. $A \cup B$ и $\neg(A \cap B)$
4. $\neg(A \cup B)$ и $A \cap B$
5. $A - B$ и $A \cap \neg B$
6. $A \setminus B$ и $A \cup B \cap \neg(A \cap B)$

Обозначения:

«/» - отрицание/дополнение (в результирующее множество попадают все элементы универсального множества, которые не вошли в множество А)

Дополнение множества это все множество всех возможных элементов (универсальное множество), за исключением тех, которые входят в множество, т.е. $\neg A = U - A$. Например, $U = \{1, 2, 3, 4, 5\}$, $A = \{2, 4\}$, $\neg A = \{1, 3, 5\}$.

«\» - симметрическая разность (в результирующее множество попадают все элементы, входящие в множества А и В, но не входящие в множество $A \cap B$).

« - » вычитание (в результате во множестве остаются элементы, которые входят в множества А, и не входят в множество В.

\cup, \cap - объединение и пересечение множеств.

5. Вывод результата

1. Элементы со n-го по m-й
2. Четные по значению элементы
3. Нечетные по значению элементы
4. Четные по индексу элементы

5. Нечетные по индексу элементы
6. Элементы, кратные H_1 или H_2 .
7. Элементы в обратном порядке
8. 30% самых больших элементов
9. 30% самых маленьких элементов

Вспомогательный материал:

Основные методы класса-контейнера `set`:

`set_union` Возвращает объединение двух множеств

`set_intersection` Возвращает пересечение двух множеств

`set_difference` Возвращает разность двух множеств (элементы одного множества, не присутствующие в другом)

`set_symmetric_difference` Возвращает разность двух множеств (элементы одного множества, не присутствующие в другом)

`erase` Удаляет элемент из множества, если он там есть. Если такого элемента нет, то с множеством ничего не происходит

`size` Возвращает количество элементов в множестве

`find` Возвращает первый же элемент, подходящий под заданный критерий или пустой элемент (NULL), если такого элемента нет

`includes` Проверяет, не является ли множество подмножеством другого

`count_if` Возвращает количество элементов, удовлетворяющих заданному условию

`swap` Обменивает элементы (сразу все) между множествами

`operator=` Копирует элементы из одного множества в другое (делает точную копию)

Операции для множеств выполняются для класса-контейнера `vector`, а затем копируются в класс-контейнер `set`.

Пример выполнения операции симметрической разности множеств:

```
set<int> s={1,2,3}, s2={3,4,5}, s_rez;  
set_symmetric_difference(s.begin(), s.end(), s2.begin(), s2.end(), inserter(s_rez,  
s_rez.begin()));  
copy(s_rez.begin(),s_rez.end(),ostream_iterator<int>(cout," "));
```

Примечание: `s_rez` – результирующее множество. Если перед выполнением операций данное множество непустое, то необходимо его очистить с помощью операции `s_rez.clear()`.

Еще один способ вывода элементов множества:

```
for(auto i: s3)
    cout<<i<<" ";
```

Вывод четных элементов множества:

```
int i=0;
for(auto is:s2)
    if((i++)%2==0)
        cout<<is;
```

Генерация множества чисел Фибоначчи:

```
set<int> S={1,2};
int N=10; // количество членов ряда
for(int i=0;i<N-2;i++) //N-2 т.к. два первых члена уже включены в мн-во S
{
    auto p=S.end();
    S.insert(*--p+*--p);
}
```

Шифры заданий (каждая цифра означает вариант параметра задания):

1. 1 2 3 1 1
2. 2 1 4 2 2
3. 3 1 4 3 3
4. 4 5 6 4 4
5. 5 2 3 5 5
6. 1 2 3 1 5
7. 2 2 4 2 4
8. 3 2 5 3 3
9. 4 2 6 4 2
10. 5 2 3 5 1
11. 5 3 4 6 6
12. 1 4 2 1 9
13. 2 5 3 2 8
14. 3 6 4 3 7
15. 4 1 5 4 6
16. 5 2 3 5 5
17. 1 3 2 6 4
18. 2 4 1 1 3
19. 3 5 6 2 2
20. 4 6 5 3 1
21. 5 1 4 4 9
22. 1 2 5 5 8

- 23. 2 3 6 6 7
- 24. 3 4 5 1 6
- 25. 4 5 4 2 5
- 26. 5 6 3 3 4
- 27. 1 3 1 4 3
- 28. 2 4 2 5 2
- 29. 3 5 3 6 1
- 30. 3 6 4 1 2