Билет №94

1.Прочтите программу и укажите, что будет выведено на экран  
void moo()  
{  
int n = 5;  
try  
{ if (n > 4) throw n;}  
catch(int a)  
{ n = 0; throw;}  
}  
int main(int argc, char\* argv[])  
{  
try  
{ moo();}  
catch(int a)  
{ cout << a << "; ";}  
cout << "o'key" << "; ";  
return 0;  
}

a) 0; o'key;

b) 5; o'key;

c) o'key

d) 5

3.Ключевое слово для обозначения секции кода, в которую может быть передано управление в том случае, если произойдет исключение

a) try

b) throw

c) catch

d) free

5.Ключевое слово для обозначения блока кода, который может генерировать исключение

a) try

b) catch

c) throw

d) free

7.Шаблоны позволяют вам давать обобщенные определения классов и функций, в смысле произвольно используемых типов

a) переменных

b) констант

c) операторов

19.Специальный инструмент контроля кода, который выявляет формальными методами анализа без выполнения тестируемой программы неверные конструкции или неверные отношения объектов программы (ошибки формального задания) –

a) CodeChecker

b) Testbed, Testbench

21.Тестирование, которое осуществляет выявление ошибок только на выполняющейся программе с помощью специальных инструментов автоматизации тестирования – Testbed или Testbench.

a) статическое

b) динамическое

22.Реализация тестирования разделяется на этапы:

a) Создание тестового набора (test suite) путем ручной разработки или автоматической генерации для конкретной среды тестирования (testing environment).

b) Исправление ошибок.

c) Оценка результатов выполнения программы на наборе тестов с целью принятия решения о продолжении или остановке тестирования.

d) Прогон программы на тестах, управляемый тестовым монитором (test monitor, test driver с получением протокола результатов тестирования (test log).

24.Пусть программа H(x:int, y:int) реализована в машине с 64 разрядными словами, тогда мощность множества тестов для исчерпывающего тестирования программы ||(X,Y)||=

a) 2\*\*64

b) 2\*\*128

25.В чём состоит назначение Оракула

a) определять, соответствуют ли выходные данные – Yв (вычисленные по входным данным – X) желаемым результатам – Y,

b) определять желаемый результат работы программы.

34.Мутационные критерии

a) используют информацию о структуре программы (критерии так называемого "белого ящика")

b) формулируются в описании требований к программному изделию (критерии так называемого "черного ящика")

c) формулируются в терминах проверки наличия заданных свойств у тестируемого приложения, средствами проверки некоторой статистической гипотезы.

d) ориентированы на проверку свойств программного изделия на основе подхода Монте-Карло.

39.В STL содержится несколько основных сущностей. Наиболее важные из них — это

a) контейнеры

b) итераторы

c) алгоритмы

d) мультиплексоры

e) конверторы

41. Объявите итератор для list<int> values;

a) list < int > values::iterator i;

b) list < int > ::iterator i;

c) iterator i;

42.list – это:

a) Последовательный контейнер

b) Ассоциативный контейнер

c) Предоставляет быструю вставку и удаление элементов в любом месте контейнера

d) Предоставляет быстрый случайный доступ к своим элементам

47. Правильный способ для добавления элементов в конец вектора(vector<int> values):

a) values[5] \= 5

b) values.add(5)

c) values.push\_back(5)

48.Объявите итератор для vector<int> values:

a) vector::iterator i

b) vector<int>::iterator i

c) iterator i

51. Правильный способ для удаления элементов из конца вектора(vector<int>values):

a) values.pop\_back()

b) values.delete()

c) values.clear()

52. Конструкция, с помощью которой можно создавать из базовых контейнеров специализированные контейнеры, называется

a) алгоритмом

b) мультиплексором

c) адаптером контейнера

d) итератором

53.Для изменения поведения алгоритма применяют

a) итераторы

b) контейнеры

c) функциональные объекты

55. Процедуры, применяемые к контейнерам для обработки их данных различными способами - это ...

a) итераторы

b) процедуры

c) алгоритмы

d) функциональные объекты

56.В STL имеется два типа ассоциативных контейнеров

a) список

b) отображения

c) двусторонняя очередь

d) множества