МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ

«КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ»

STD и другие возможности С++

**Методические указания**

**к лабораторной работе № 7**

**по дисциплине «ООП»**

Киев 2016

СодЕржание

[1 Цель лабораторной работы 3](#_Toc469358558)

[2 Теоретические положения 4](#_Toc469358559)

[2.1. Стандартная библиотека языка C++ 4](#_Toc469358560)

[2.2. Компоненты STL 5](#_Toc469358561)

[2.3. Вектор 6](#_Toc469358562)

[2.3.1. Применение алгоритмов к вектору 9](#_Toc469358563)

[2.3.2. Создание класса в векторе 9](#_Toc469358564)

[2.4. Списки 10](#_Toc469358565)

[2.5. std::tuple 12](#_Toc469358566)

[2.6. Алгоритмы 12](#_Toc469358567)

[2.7. Генераторы случайных чисел, библиотека <random> 13](#_Toc469358568)

[2.8. Smart Pointers (Умные указатели) 17](#_Toc469358569)

[2.8.1. std::unique\_ptr 17](#_Toc469358570)

[2.8.2. std::shared\_ptr 19](#_Toc469358571)

[2.8.3. std::weak\_ptr 20](#_Toc469358572)

[2.9. Лямбда-выражения в C++ 22](#_Toc469358573)

[3 Задания 24](#_Toc469358574)

[4 Требования к отчету 53](#_Toc469358575)

[5 Контрольные вопросы 54](#_Toc469358576)

# Цель лабораторной работы

Цель работы – изучить особенности работы со стандартной библиотекой STD.

# Теоретические положения

## Стандартная библиотека языка C++

В языке программирования C++ термин **Стандартная Библиотека** означает коллекцию классов и функций, написанных на базовом языке. Стандартная Библиотека поддерживает несколько основных контейнеров, функций для работы с этими контейнерами, объектов-функции, основных типов строк и потоков (включая интерактивный и файловый ввод-вывод), поддержку некоторых языковых особенностей, и часто используемые функции для выполнения таких задач, как, например, нахождение квадратного корня числа. Стандартная Библиотека языка C++ также включает в себя спецификации стандарта ISO C90 стандартной библиотеки языка Си. Функциональные особенности Стандартной Библиотеки объявляются внутри пространства имен *std*.

Стандартная библиотека шаблонов (STL) – подмножество стандартной библиотеки C++ и содержит контейнеры, алгоритмы, итераторы, объекты-функции и т.д. Хотя некоторые программисты используют термин «STL» вместе (или попеременно) с термином «Стандартная библиотека C++».

Заголовочные файлы стандартной библиотеки C++ не имеют расширения «.h».

Стандартная библиотека C++ содержит последние расширения C++ стандарта ANSI (включая библиотеку стандартных шаблонов и новую библиотеку iostream). Она представляет собой набор файлов заголовков. В новых файлах заголовков отсутствует расширение h.

## Компоненты STL

В **STL** большое количество шаблонов, как классов так и функций. Мы можем их использовать с ООП или без него. В **STL** есть 3 основные компоненты.

* Итераторы
* Контейнеры
* Алгоритмы

***Итератор*** – это аналог указателя, с помощью можно получать доступ к различных элементам данных. Можно использовать и пару итераторов для задания диапазона. Как и указатель для получения данных из итераторов их необходимо разыменовать с помощью операции \*. Всего есть пять классов итераторов.

|  |  |
| --- | --- |
| Итератор произвольного доступа | Хранит и извлекает значения. Обеспечивает произвольный доступ к элементам |
| Двунаправленный итератор | Хранит и извлекает значения. Перемещается вперёд и назад |
| Прямой итератор | Хранит и извлекает значения. Перемещается только вперёд |
| Итератор ввода | Извлекает, но не хранит элементы. Перемещается только вперёд |
| Итератор вывода | Хранит, но не извлекает значения. Перемается только вперёд. |

***Контейнеры –*** это структуры данных, такие как списки, очереди и так далее. Доступ к данным находящимся внутри контейнера осуществляется с помощью итераторов. Выделяют следующие контейнеры:

vector – линейный массив;

list – двухсвязанный список;

deque – очередь с двухсторонним доступом;

set – ассоциативный массив уникальных ключей;

multiset – ассоциативный массив с возможность дублирования ключей;

map – ассоциативный массив с уникальными ключами и значениями;

multimap –ассоциативный массив с возможность дублирования ключей и значений;

stack – структура данных типа стек;

queue – структура данных типа очередь.

***Алгоритмы*** – это шаблоны функций, с помощью которых производятся операции по работе с данными.

## Вектор

Вектор (vector) напоминает массив, обладающий возможностью расти до произвольного размера, и поддерживать информацию о размере. Аналогично массиву к вектору можно обратиться с использованием операцией индексирования []. Основные характеристики вектора:

* доступ к данным с одинаковой скоростью;
* вставка приводит к перемещению элементов;
* при расширении данные копируются в другой блок

Вектор оптимален для получения информации, но при большом количестве вставок лучше воспользоваться другими контейнерами, например, списками. Проблема в том, что физически вектор располагается в непрерывной памяти. На C это реализовывали функциями malloc например.

#include "vector" //подключение заголовочного файла

using namespace std; //объявление рабочей памяти

vector< int > vArray1; //или vector< int > vArray2(30);

Можно получать информацию о параметрах вектора.

**size()** - сколько данных храниться

**capacity()** – сколько элементов может храниться до изменения размера

**max\_size()** – максимальный размер обычно равен наиболее большому доступному блоку памяти

Например:

#include "conio.h"

#include "iostream.h"

#include "vector.h"

#include "iostream.h"

using namespace std;

void main() {

vector <int> vArray2(30);

cout << "Size Vector " << vArray2.size() << endl;

cout << "Capacity Vector " << vArray2.capacity() << endl;

cout << "Max\_Size Vector " << vArray2.max\_size() << endl;

for (int x=1;x<5;x++) vArray2.push\_back(10);

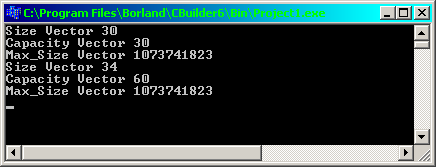
cout << "Size Vector " << vArray2.size() << endl;

cout << "Capacity Vector " << vArray2.capacity() << endl;

cout << "Max\_Size Vector " << vArray2.max\_size() << endl;

getch();

}



Как видите **Size** показывает сколько сейчас лежит в векторе чисел. В то время как **capacity** возвращает инициализированный размер, то есть тот размер, до которого можно добавлять данные без инициализации.

Вектор можно инициализировать с заранее установленными значениями. Вот пример демонстрирующий доступ к данным вектора через **[]**.

vector <int> vVec(5,10);

for (int x=0;x < 5;x++) cout << vVec[x] << endl;

Рассмотрим некоторые функции вектора:

1. **assign** – заполнение части вектора, например:

vector <int> vVec(5,10);

vVec.assign(3,2);

for (int x=0;x < 5;x++) cout << vVec[x] << endl;

2. **front()** и **back()** – получение первого и последнего элемента, например:

vector <int> vVec(5,10);

vVec.assign(5,1);

vVec[0]=0;

vVec[4]=4;

cout << vVec.front() << " " << vVec.back() << endl;

3. **insert** – вставка элемента с перемещением. Вставка производится в первую позицию с перемещением элементов вниз.

vector <int> vVec(5,10);

vVec.assign(5,1);

vVec[0]=0;

vVec[4]=4;

for (int x=0;x < 5;x++) cout << vVec[x] << " ";

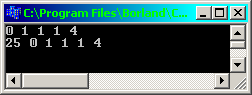
cout << endl;

vVec.insert(vVec.begin(),25);

for (int x=0;x < 6;x++) cout << vVec[x] << " ";

cout << endl;

getch();



4. **push\_back()** – помещение числа в конец вектора:

vector <int> vVec(5,10);

vVec.assign(5,1);

vVec[0]=0;

vVec[4]=4;

for (int x=0;x < 5;x++) cout << vVec[x] << " ";

cout << endl;

vVec.insert(vVec.begin(),25);

for (int x=0;x < 6;x++) cout << vVec[x] << " ";

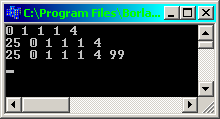
cout << endl;

vVec.push\_back(99);

for (int x=0;x < 7;x++) cout << vVec[x] << " ";

cout << endl;

getch();



5. **pop\_back()** – удаление последнего элемента с сокращением размера.

vVec.pop\_back();

for (int x=0;x < vVec.size();x++) cout << vVec[x] << " ";

6. **erase()** – удаление элемента

vVec.erase(vVec.begin()+2,vVec.begin()+4 );

for (int x=0;x < vVec.size();x++) cout << vVec[x] << " ";

7. **resize()** – изменение размера вектора функция **resize**

vVec.resize(3);

for (int x=0;x < vVec.size();x++) cout << vVec[x] << " ";

cout << endl;

### Применение алгоритмов к вектору

Одним из алгоритмов вектора является сортировка, для неё можно применить стандартный алгоритм **sort**. Для его использования необходимо подключить файл заголовков алгоритмов.

#include "conio.h"

#include "iostream.h"

#include "vector.h"

#include "iostream.h"

#include "algorithm.h"

using namespace std;

void main() {

vector< int > v1(10);

vector< int > v2(10);

int x;

for (x=0;x < v1.capacity() ;x++) v1[x]=10-x;

for (x=0;x < v1.capacity() ;x++) v2[x]=10-x;

for (x=0;x < v1.size();x++) cout << v1[x] << " ";

cout << endl;

for (x=0;x < v2.size();x++) cout << v2[x] << " ";

cout << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_ SORT \_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

sort(v1.begin(),v1.end());

for (x=0;x < v1.size();x++) cout << v1[x] << " ";

cout << endl;

sort(v2.begin()+1,v2.end()-1);

for (x=0;x < v2.size();x++) cout << v2[x] << " ";

cout << endl;

getch();

}



### Создание класса в векторе

Для того чтобы поместить вектор в класс необходимо в классе задать:

* Конструктор по умолчанию.
* Конструктор копий.
* Деструктор.

Например:

#include <vector.h>

#include <iostream.h>

#include <conio.h>

using namespace std;

class TMyClass {

public:

TMyClass():x(0),y(0){}; // конструктор по умолчанию

TMyClass(const TMyClass &my); // конструктор копий

TMyClass(int xx,int yy):x(xx),y(yy){};

~TMyClass(); // деструктор

int x;

int y;

};

TMyClass::TMyClass(const TMyClass &my){

x=my.x;

y=my.y;

}

TMyClass::~TMyClass(){}

void main() {

vector<TMyClass> v;

v.push\_back(TMyClass(1,1));

v.push\_back(TMyClass(2,2));

v.push\_back(TMyClass(3,3));

for (int x=0; x<v.size();x++)

cout << v[x].x << " " << v[x].y << endl;

getch();

};

## Списки

Для использования списков необходимо подключить заголовочный файл и выбрать область.

#include "list.h"

using namespace std;

Теперь список можно объявлять.

list< int > intList; //пустой список

list< int > intListTest(3); //список на 3 элемента

list< int > intListInit(3,1); //список на три элемента, проиниц. числом 1

Количество элементов определяем функцией **size**

cout << intListInit.size() << endl;

Можно проверить список на пустоту.

if (intList.empty()) cout << " int List empty " << endl;

1. Для вставки можно использовать три метода **insert**, **push\_back**, **push\_front**.

#include "iostream"

#include "list"

using namespace std;

list< int > intList;

list< int > intListTest(3);

list< int > intListInit(3,1);

void main(){

cout << intListInit.size() << endl;

if (intList.empty()) cout << " int List empty " << endl;

intListInit.push\_back(100);

intListInit.push\_front(0);

intListInit.insert(intListInit.begin(),55);

}

2. Сортировку списка можно произвести при помощи функции **sort()**.

#include <iostream.h>

#include <list.h>

#include <stdlib.h>

#include <conio.h>

using namespace std;

list <int> lst;

void output\_list(){

list <int> ::iterator p = lst.begin();

while(p != lst.end()){ cout <<\*p<<" "; p++;}

}

void main(){

int i;

for (i=0; i<10; i++) lst.push\_back(rand());

cout<<"List is: \n";

output\_list();

cout<<endl;

lst.sort();

cout <<"Sorted list is: \n";

output\_list();

getch();

}

2. Из списка можно удалять элементы равные определенному значению.

intListInit.remove(1);

3. Два вектора можно объединить , вызвав функцию **splice()**. С помощью функции **merge()** один список можно внедрить в другой (для упорядоченных списков).

#include <iostream.h>

#include <list.h>

#include <stdlib.h>

#include <conio.h>

using namespace std;

list <int> lst1, lst2;

void output\_list(list <int> lst){

list <int> ::iterator p = lst.begin();

while(p != lst.end()){ cout <<\*p<<" "; p++; }

cout<<endl;

}

void main(){

int i;

for (i=0; i<10; i+=2) lst1.push\_back(i);

for (i=1; i<11; i+=2) lst2.push\_back(i);

cout<<"Lists are: \n";

output\_list(lst1);

output\_list(lst2);

lst1.merge(lst2);

cout <<"Merged list is: \n";

output\_list(lst1);

getch();

}

## std::tuple

Шаблонный класс std::tuple - коллекция фиксированного размера с гетерогенными значениями (кортеж). Это обобщение std::pair.

Определено в заголовочном файле <tuple>

std::tuple<std::string, int, double> tuple\_("hello", 42, 3.14);

## Алгоритмы

В библиотеке STL определено большое количество алгоритмов, представляющих собой шаблонные функции. Рассмотри некоторые из них:

|  |  |
| --- | --- |
| **Алгоритм** | **Описание** |
| binary\_search | Выполняет бинарный поиск в упорядоченной последовательности |
| copy | Копирует последовательность |
| copy\_backward | Копирует последовательность, начиная с последнего элемента |
| count | Возвращает число элементов последовательности |
| count\_if | Возвращает число элементов последовательности, удовлетворяющих определённому условию |
| equal | Определяет, совпадают ли элементы двух диапазонов |
| fill | Заполняет диапазон значениями |
| find | Находит диапазон, содержащий заданное значение |
| for\_each | Применяет функцию к диапазону элементов |
| max\_element | Возвращает итератор, установленный на максимальный элемент диапазона |
| min\_element | Возвращает итератор, установленный на минимальный элемент диапазона |
| reverse | Меняет порядок следования элементов диапазона на противоположный |
| remove | Удаляет элементы из указанного диапазона |
| sort | Упорядочивает диапазон |
| swap | Меняте местами два элемента |
|  |  |

## Генераторы случайных чисел, библиотека <random>

**Генератор случайных чисел**– это объект, формирующий последовательность из псевдослучайных чисел.

В языке С++ для генерации разного рода псевдослучайных последовательностей рекомендуется использовать библиотеку <random>.

#include <random>

В отличии от устаревшей функции языка С – rand, библиотека <random> позволяет генерировать, в зависимости от требований, последовательности псевдослучайных чисел, распределенных по одному из известных законов.

Библиотека позволяет генерировать случайные числа, используя комбинации генераторов и распределений.

Генераторы: объекты, которые генерируют равномерно распределенные числа.

Распределения: объекты, которые преобразуют последовательности чисел, генерируемых с помощью генератора, в последовательности чисел, распределенных в соответствии с заданным законом распределения, например равномерный, нормальный, биномиальный и т.д.

Фактически «распределения» создают случайную величину с заданным законом распределения.

Рассмотрим одно из сочетаний «генератора» и «распределения», которое можно использовать в большинстве приложений, где требуются случайные числовые последовательности.

Генератор: std::mt19937.

typedef mersenne\_twister\_engine<uint\_fast32\_t,32,624,397,31,0x9908b0df,11 ,0xffffffff,7,0x9d2c5680,15,0xefc60000,18,1812433253> mt19937;

Генератор псевдослучайных чисел, реализующий генератор «Вихрь Мерсена».

**Вихрь Мерсенна (Mersenne twister)** генератор псевдослучайных чисел (ГПСЧ), разработанный в 1997 году японскими учёными Макото Мацумото и Такудзи Нисимура. Вихрь Мерсенна основывается на свойствах простых чисел Мерсенна (отсюда название) и обеспечивает быструю генерацию высококачественных по критерию случайности псевдослучайных чисел.

Вихрь Мерсенна лишен многих недостатков, присущих другим ГПСЧ, таких как малый период, предсказуемость, легко выявляемые статистические закономерности.

Тем не менее, этот генератор не является криптостойким, что ограничивает его использование в криптографии.

Пользовательские функции генератора mt19937 приведены в таблице 3.1.

Таблица 3.1 – Пользовательские функции генератора

|  |  |
| --- | --- |
| (constructor) | Конструктор, в качестве параметра можно задать начальное значение генератора |
| min | Минимальное число, которое может быть сгенерировано |
| max | Максимальное число, которое может быть сгенерировано |
| seed | Функция, отвечающая за задание начального числа генератора |
| operator() | Генерирует случайное число |
| discard | Изменяет внутренне состояние генератора |

Для более качественной генерации чисел начальное значение генератора следует задавать всегда разным случайным числом. Можно использовать функцию time с параметром 0 (time(0)), а можно использовать std::random\_device – генератор истинных случайных чисел. Его оператор «()» возвращает случайное число.

std::random\_device rd;

cout << rd() << endl;

Распределение: std::uniform\_int\_distribution

template <class IntType = int> class uniform\_int\_distribution;

Реализует равномерное дискретное распределение псевдослучайных чисел.

**Пример 1**

#include "stdafx.h"

#include <random>

#include <iostream>

using namespace std;

int main()

{

random\_device rd;

mt19937 gen(rd());

uniform\_int\_distribution<int> dist(1, 6);

for (int i = 0; i < 5; i++) {

cout << dist(gen) << " ";

}

cout << rd() << endl;

system("pause");

return 0;

}

Данная программа генерирует псевдослучайные числа в диапазоне от 1 до 6, при этом вероятность выпадения каждого числа одинакова.

Генераторы псевдослучайных чисел имеют широчайшую область применения в прикладных задачах программирования, рассмотрим пример генерации тестовых наборов данных.

**Пример 2**

#include "stdafx.h"

#include <algorithm>

#include <array>

#include <iostream>

#include <random>

#include <string>

#include <vector>

using namespace std;

template <class type> void print(const type& t) { //шаблонная функция вывода

for (const auto& tt : t) {

cout << tt << " ";

}

cout << endl;

}

int main()

{

random\_device rd; // "истинно" случайное число, начальное значение генератора

mt19937 gen(rd()); // генератор Вихрь Мерсена

uniform\_int\_distribution<int> dist(0, 5); // равномерное дискретное распределение

vector<int> v; //вектор

for (int i = 0; i < 6; ++i) { // заполняем вектор случайными числами

v.push\_back(dist(gen));

}

cout << "Randomized vector: ";

print(v);

array<string, 6> arr = { { "AAA", "BBB", "CCC", "DDD", "EEE", "FFF" } }; // массив слов

shuffle(arr.begin(), arr.end(), gen); //случайным образом "перемешиваем" массив слов

cout << "Randomized array: ";

print(arr);

cout << "Randomized data set: " << endl; //выводим наш набор

for (int i = 0; i < 6; ++i) {

cout << v[i] << " : " << arr[i] << endl;

}

system("pause");

return 0;

}

## Smart Pointers (Умные указатели)

**Smart pointer** – это объект, работать с которым можно как с обычным указателем, но при этом, в отличии от последнего, он предоставляет некоторый дополнительный функционал (например, автоматическое освобождение закрепленной за указателем области памяти).

Умные указатели призваны для борьбы с утечками памяти, которые сложно избежать в больших проектах. Они особенно удобны в местах, где возникают исключения, так как при последних происходит процесс раскрутки стека и уничтожаются локальные объекты. В случае обычного указателя – уничтожится переменная-указатель, при этом ресурс останется не освобожденным. В случае умного указателя – вызовется деструктор, который и освободит выделенный ресурс.

В новом стандарте появились следующие умные указатели: unique\_ptr, shared\_ptr и weak\_ptr. Все они объявлены в заголовочном файле <memory>.

### std::unique\_ptr

Прототип:

template<class T, class Deleter = std::default\_delete<T>> class unique\_ptr;

template <class T, class Deleter > class unique\_ptr<T[],Deleter>;

std::unique\_ptr – умный указатель, который:

* получает единоличное владение объектом через его указатель, и
* разрушает объект через его указатель, когда unique\_ptr выходит из области видимости.

unique\_ptr не может быть скопирован или задан через операцию присвоения, два экземпляра unique\_ptr не могут управлять одним и тем же объектом. Неконстантный unique\_ptr может передать владение управляемым объектом другому указателю unique\_ptr. const std::unique\_ptr не может быть передан, ограничивая время жизни управляемого объекта областью, в которой указатель был создан. Когда unique\_ptr уничтожается, он удаляет объект с помощью Deleter.

Существует две версии std::unique\_ptr:

1) управляет временем жизни одного объекта, например, созданного с помощью оператора new

2) управляет временем жизни массива, с длиной, определенной во время выполнения, созданного с помощью new[]

Типичные случаи применения std::unique\_ptr включают:

* обеспечение безопасности исключений для классов и функций, которые управляют объектами с динамическим временем жизни, гарантируя удаление в случае нормального завершения и завершения по исключению
* передача владения динамически созданным объектом в функции
* получение владения динамически созданным объектом из функций
* в качестве типа элемента в контейнерах, поддерживающих семантику перемещения, таких как std::vector, которые хранят указатели на динамически выделенные объекты (например, если желательно полиморфное поведение)

**Пример 1**

#include "stdafx.h"

#include <iostream>

#include <memory>

using namespace std;

class Point

{

private:

int x, y;

public:

Point(){}

Point(int px, int py) : x(px), y(py){ }

void SetPoint(int px, int py){ x = px; y = py; }

void Print(){ std::cout << x << " " << y << endl; }

};

int main()

{

unique\_ptr<Point> A(new Point(42,42));

A->Print();

system("pause");

return 0;

}

### std::shared\_ptr

Прототип:

template< class T > class shared\_ptr;

std::shared\_ptr – умный указатель, с разделяемым владением объектом через его указатель. Несколько указателей shared\_ptr могут владеть одним и тем же объектом; объект будет уничтожен, когда последний shared\_ptr, указывающий на него, будет уничтожен или сброшен. Объект уничтожается с использованием delete-expression или с использованием пользовательской функции удаления объекта, переданной в конструктор shared\_ptr.

shared\_ptr может не владеть ни одним объектом, в этом случае он называется пустым.

shared\_ptr отвечает требованиям CopyConstructible и CopyAssignable.

**Пример 2**

#include "stdafx.h"

#include <iostream>

#include <memory>

using namespace std;

class Point

{

private:

int x, y;

public:

Point(){}

Point(int px, int py) : x(px), y(py){ }

void SetPoint(int px, int py){ x = px; y = py; }

void Print(){ std::cout << x << " " << y << endl; }

};

int main()

{

shared\_ptr<Point> A(new Point(42,42));

shared\_ptr<Point> B(new Point);

B = A;

B->Print();

system("pause");

return 0;

}

### std::weak\_ptr

Прототип:

template< class T > class weak\_ptr;

std::weak\_ptr – умный указатель, который содержит "слабую" ссылку на объект, управляемый указателем std::shared\_ptr. Он должен быть преобразован в std::shared\_ptr, чтобы получить доступ к управляемому объекту.

std::weak\_ptr моделирует временное владение: когда объект должен быть доступен только если он существует и может быть удален в любой момент кем-то другим, std::weak\_ptr используется для отслеживания объекта, и преобразуется в std::shared\_ptr для принятия временного владения. Если исходный std::shared\_ptr будет уничтожен в процессе работы, время жизни объекта продлевается до того момента, пока не будет разрушен временный std::shared\_ptr.

Метод lock создает shared\_ptr, который управляет объектом, на который ссылается weak\_ptr.

Помимо этого, std::weak\_ptr используется для устранения циклических ссылок std::shared\_ptr.

**Пример 3**

#include "stdafx.h"

#include <iostream>

#include <memory>

using namespace std;

class Point

{

private:

int x, y;

public:

Point(){}

Point(int px, int py) : x(px), y(py){ }

void SetPoint(int px, int py){ x = px; y = py; }

void Print(){ cout << x << " " << y << endl; }

};

void out(weak\_ptr<Point> Twp)

{

if (auto spt = Twp.lock()) { // необходимо скопировать в shared\_ptr перед использованием

spt->Print();

}

else {

cout << "weak\_ptr\_point is expired\n";

}

}

int main()

{

weak\_ptr<Point> weak\_ptr\_point;

{

auto sp = std::make\_shared<Point>(42, 42);

weak\_ptr\_point = sp;

out(weak\_ptr\_point);

}

out(weak\_ptr\_point);

system("pause");

return 0;

}

## Лямбда-выражения в C++

Лямбда-выражение (или просто лямбда) в C++11 – это удобный способ определения анонимного объекта-функции непосредственно в месте его вызова или передачи в функцию в качестве аргумента. Обычно лямбда-выражения используются для инкапсуляции нескольких строк кода, передаваемых алгоритмам или асинхронным методам.

В стандарте ISO C++ демонстрируется простое лямбда – выражение, передаваемое функции std::sort() в качестве третьего аргумента :

#include "stdafx.h"

#include <iostream>

#include <algorithm>

using namespace std;

void abssort(float\* x, unsigned n) {

sort(x, x + n,

// Lambda expression begins

[](float a, float b) {

return (abs(a) < abs(b));

} // end of lambda expression

);

}

int main()

{

float arr[] = {3.5,-1,6,-7};

abssort(arr, 4);

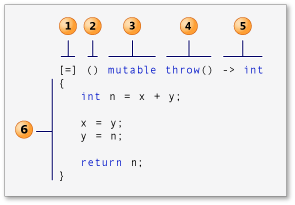
for (int i = 0; i < 4; i++) cout << arr[i] << endl;

system("pause");

return 0;

}

На следующем рисунке показана структура лямбда-выражения:



(1) предложение фиксации (в спецификации C++ это lambda-introducer.)

(2) список параметров (необязательно). (Также именуется как lambda declarator.)

(3) отключаемая спецификация (необязательно).

(4) спецификация исключений (необязательно).

(5) завершающий возвращаемый тип (необязательно).

(6) тело лямбда-выражения.

Пример

#include <iostream>

using namespace std;

int main()

{

int m = 0;

int n = 0;

[&, n] (int a) mutable { m = ++n + a; }(4);

cout << m << endl << n << endl;

}

# Задания

Разработать пошаговый алгоритм решения расчётной части задачи и подзадач (при необходимости). Разработать UML диаграмму классов. Выполнить программную реализацию задания согласно варианту. Прототипы классов должны содержаться в отдельном \*.h-файле. В программе обязательно предусмотреть вывод информации об исполнителе работы (ФИО), номере варианта, выбранном уровне сложности и задании. Предусмотреть возможность повторного запуска программы (запрос о желании выйти из программы, или продолжить работу). Если программная реализация предполагает решение нескольких подзадач, разработать меню для демонстрации каждой из них. Ключевые моменты программы обязательно должны содержать комментарии. При выполнении лабораторной работы для объявления и обработки структур данных использовать std контейнеры и алгоритмы.

**Уровень А (1 балл)**

1. Построить список, элементами которого являются слова текста. Определить, является ли симметричным введённый текст.

2. Задана последовательность слов из латинских букв, слова разделены пробелами. После последнего слова последовательности расположена точка. Вывести на экран слова по алфавиту, для каждого слова определить число его вхождений в последовательность.

3. Задана последовательность слов. Необходимо построить список, в котором каждое следующее слово начинается на букву, на которую заканчивается предыдущее слово. Если такой список сделать нельзя, то построить список, в котором слова расположены по алфавиту.

4. Задан текст, состоящий из нескольких строк. Построить стек, элементами которого являются буквы. Напечатать текст, в котором буквы каждой строки отображаются в обратном порядке.

5. Построить список, элементами которого являются слова текста. Слова разделяются запятыми, после последнего слова стоит точка. Напечатать все слова заданной длины (желательно использовать индексный список).

6. Задано натуральное число n и действительные числа X1, X2, … , Xn. Определить произведение вида X1⋅Xn + X2⋅Xn-1 + … . Использовать двухсвязный список.

7. Построить односвязный список и заполнить его. Определить длину списка. Найти в списке узел с заданным значением, напечатать его порядковый номер.

8. Построить односвязный циклический список. Удалить из списка узел, номер которого введён пользователем. Напечатать входной и выходной списки.

9. Задано натуральное число n и действительные числа X1, X2, … , Xn. Получить последовательность вида X1–Xn , X2­–Xn , … . Использовать циклический список.

10. Задана последовательность целых чисел. Построить список, в котором числа упорядочены по условию возрастания. Напечатать упорядоченный список. Добавить с клавиатуры новое число. Найти его место в списке, вывести новый список.

11.Задана последовательность разных натуральных чисел. Построить список, элементами которого являются натуральные числа. Конец списка – число 0. Напечатать в обратной последовательности все числа последовательности.

12. Ввести две последовательности слов L1 и L2. Создать список L, в котором содержатся слова, входящие в оба списка.

13. Создать циклический список, элементами которого являются числа. Последовательно удалить каждое третье число, подсчитать число удалений. Вывести начальный список и элементы, которые были удалены.

14. Создать двухсвязный список, элементами которого являются слова текста. Вывести слова, расположенные на чётных позициях при проходе по списку в одном направлении, слова на нечётных – при проходе в обратном направлении.

15. Создать список, элементы которого представляют собой отсортированную по убыванию последовательность. Ввести число с клавиатуры, найти его место в списке. Вывести полученный список.

16. Задан список, элементами которого являются различные символы. Необходимо в конец списка вставить новый элемент, введённый с клавиатуры, и удалить первый элемент. Напечатать исходный и конечный списки.

17. Задано две последовательности – последовательность целых чисел и последовательность действительных чисел. Построить список, в котором числа этих двух последовательностей чередуются. Вывести конечный список, и отдельно его подсписки.

18. Создать двухсвязный список. Ввести с клавиатуры новый элемент и число k, вставить новый элемент на k-ю позицию списка. Разделить список на два подсписка, сделав введённый элемент последним в первом подсписке и первым во втором. Вывести конечные подсписки.

19. Задан список, элементами которого являются различные символы. Необходимо в начало списка вставить новый элемент, введённый с клавиатуры, и удалить последний элемент. Напечатать исходный и конечный списки.

20. Задан текст. Необходимо построить список слов текста. Определить, сколько раз встречается каждое слово в тексте. Вывести результаты на экран.

21. Задан текст. Необходимо построить список слов текста. Упорядочить список по условию убывания длины слова. Вывести на экран полученный список.

22. Ввести две последовательности действительных чисел L1 и L2. Создать список L, в котором содержаться все элементы списков L1 и L2, упорядоченные по возрастанию.

23. Задана последовательность целых чисел. Построить список, в котором числа упорядочены по возрастанию. Исключить из списка все отрицательные элементы. Вывести на экран исходный и конечный списки.

24. Задан текст с равным количеством открываемых и закрываемых скобок. Построить список, элементами которого являются символы, находящиеся между парой скобок. Напечатать номера позиций в тексте каждой пары скобок.

25. Создать список. Ввести с клавиатуры число k и новый элемент списка. Удалить элемент, стоящий перед k-м, новый элемент добавить после k-го. Вывести начальный и конечный списки.

26. Ввести две последовательности целых чисел L1 и L2. Создать список L, в котором содержаться элементы списка L1, не входящие в список L2.

27. Задано натуральное число n и действительные числа X1, X2, … , Xn. Определить произведение вида (X1–Xn)(X2 –Xn-1) … (Xn–X1). Использовать двухсвязный список.

28. Построить список, элементами которого являются натуральные числа. Определить, является ли список симметричным.

29. Создать двухсвязный список, элементами которого являются действительные числа. Подсчитать сумму элементов, расположенных на чётных позициях, при прохождении списка в одну сторону; сумму элементов, расположенных на нечётных позициях – при проходе списка в другую сторону. Результаты вывести на экран.

30. Ввести две последовательности чисел L1 и L2. Ввести произвольное число, проверить, принадлежит ли оно одновременно обоим последовательностям.

**Уровень Б (+1 балл)**

Создать шаблон класса, в котором содержится массив объектов заданного типа, помимо базовых типов, в шаблоне необходимо предусмотреть использование пользовательских типов заданных в таблице 3.2. Для пользовательских типов перегрузить все необходимые для манипуляций операции.

Разработать для данного класса алгоритмы сортировки согласно варианту (таблица 3.2).

Продемонстрировать все действия с объектом данного класса, на нескольких базовых и всех пользовательских типах данных (обязательно тип **char \***).

Корректная сортировка для типа int значений 5, 10, 9, 101, будет иметь вид: 5, 9, 10, 101.

Корректная сортировка для типа char \* значений “5”, “10”, “9”, “101”, будет иметь вид: “10”, “101”, “5”, “9”.

Таблица 3.2 – Варианты заданий (уровень Б)

|  |  |  |
| --- | --- | --- |
| **№** | **Пользовательские типы** | **Метод сортировки** |
| 1 | string, fraction | Шейкер-сортировка |
| 2 | date, binary number | Метод Шелла |
| 3 | time, fraction | Сортировка слиянием |
| 4 | fraction, date | Быстрая сортировка |
| 5 | date-time | Пирамидальная сортировка |
| 6 | set (template) | Сортировка вставками |
| 7 | binary number, fraction | Интроспективная сортировка |
| 8 | set (template) | Плавная сортировка |
| 9 | hex number, string | Сортировка выбором |
| 10 | string, time | Timsort |
| 11 | date, fraction | Шейкер-сортировка |
| 12 | date-time | Метод Шелла |
| 13 | fraction, string | Сортировка слиянием |
| 14 | time, date | Быстрая сортировка |
| 15 | set (template) | Пирамидальная сортировка |
| 16 | binary number, time | Сортировка вставками |
| 17 | oct number, date | Интроспективная сортировка |
| 18 | hex number, time | Плавная сортировка |
| 19 | string, date | Сортировка выбором |
| 20 | set (template) | Timsort |
| 21 | time, fraction | Шейкер-сортировка |
| 22 | fraction, oct number | Метод Шелла |
| 23 | date-time | Сортировка слиянием |
| 24 | set (template) | Быстрая сортировка |
| 25 | binary number, string | Пирамидальная сортировка |
| 26 | oct number, fraction | Сортировка вставками |
| 27 | hex number, fraction | Интроспективная сортировка |
| 28 | date, fraction | Плавная сортировка |
| 29 | time, date | Сортировка выбором |
| 30 | set (template) | Timsort |

**Уровень В (+2 балла)**

Согласно варианту создать набор связанных классов для хранения информации о предметной области. Предусмотреть в них необходимые атрибуты, конструкторы, методы и т.д.

Создать коллекцию данных предметной области и выполнить задание согласно варианту.

Данные о предметной области должны генерироваться случайным образом.

Для создания динамических объектов использовать Smart pointers.

1. **Детский садик**. В детский садик поступают запросы на размещение детей согласно их возрасту по различным группам (ясли, младшая, средняя и т.д.). Заведующая садика рассматривает запрос, и в случае возможности его выполнения резервирует место за ребёнком на две недели, в течении которых необходимо предоставить медицинские справки о здоровье ребёнка.

Для получения медицинских справок необходимо предоставить в поликлинику сведения о детском садике, в котором планируется отправить ребёнка, и получить карточку для прохождения медосмотра. После прохождения медосмотра выдаётся справка о здоровье ребёнка, по результатам которой ребёнок зачисляется в сад.

Если получение медицинской справки затягивается более чем на две недели, то заведующая детским садиком освобождает зарезервированное место.

Численность группы строго ограничена.

Сформировать коллекции данных о комплектации групп.

**Задание.**

Определить количество занятых, свободных и зарезервированных мест по каждой группе.

2. **Товары в кредит**. Мебельный магазин предоставляет услуги по приобретению товаров в кредит. Для оформления кредита покупатель должен предоставить справку о своих доходах. Решение о выдаче кредита осуществляется путём сопоставления цены товара и доходов покупателя: определяется срок погашения кредита, процентная ставка кредита. На основании этих данных формируется график погашения кредита, определяющий дату и сумму очередного взноса. Сформированный график передаётся администрации магазина.

Если клиент не выплачивает взнос в установленное время, то насчитывается пеня за каждый день просрочки. Пока пеня не погашена, выплата взносов за последующие месяцы не принимается.

Оформление договора на кредит происходит при участии определённого банка, за что банк получает фиксированный процент комиссии. Выбор банка производит администрация мебельного магазина.

Сформировать коллекцию данных о клиентах мебельного магазина с полной информацией о них и приобретенных в кредит товарах.

**Задание.**

Определить количество клиентов, просрочивших выплату, и сумму задолженности по каждому из них.

3. **Обмен валют**. Процесс обмена денег в пункте валют происходит следующим образом: клиент запрашивает кассира о возможности перевода определённой суммы денег из одной валюты в другую. Кассир проверяет имеющуюся наличность в соответствующей валюте и сообщает клиенту о возможности или невозможности выполнения операции. При проведении операции формируется квитанция, в которой указываются названия валют, курс перевода, сумма и дата операции. При выдаче денег клиент дополнительно может попросить выдать ему сумму определёнными купюрами, или осуществить частичный размен и обмен денег (к примеру, осуществить размен 100 у.е. на 50 у.е и соответствующий остаток в национальной валюте).

В конце рабочего дня кассир составляет сведенный отчёт, в котором указывается сумма каждой валюты на начало рабочего дня и к его концу, а также прибыль, полученная за день; сформированный отчёт передаётся руководству обменного пункта. Если резерв какой-то валюты достигает критического минимума, то кассир также сообщает об этом своему руководству.

Сформировать коллекцию данных обо всех валютных операциях за день.

**Задание.**

Определить общую сумму продажи и покупки каждой валюты.

4. **Фотостудия**. Работа фотостудии осуществляется следующим образом – клиент оформляет заказ (печать фотографий, проявка плёнки) у приёмщика. При оформлении заказа приёмщик записывает реквизиты клиента (фамилия) и желаемый срок выполнения заказа. Заказ оформляется на двух бланках, первый из которых остаётся у клиента, а второй передаётся специалисту-фотографу. В зависимости от скорости выполнения заказа определяется цена: срочные заказы имеют наценку в 25% от стоимости заказа. Оплата заказа осуществляется после его реализации.

Приёмщик в течении дня передаёт бланки с заказами фотографу, который выполняет их и передаёт обратно приёмщику. При выполнении заказа фотограф использует определённые фотоматериалы (бумага, проявитель и т.д.), учёт которых производится администратором салона.

Сформировать коллекцию данных обо всех заказах, с видом заказа, объемом заказа, выручкой и расходами.

**Задание.**

Определить вид заказа, число заказов данного вида и сумма выручки по каждому виду заказов.

5. **Детский садик**. В детском саду проводятся развлекательные мероприятия: театральные выступления, экскурсии, походы в цирк и т.д. Организаторы развлечений являются постоянными партнёрами садика, и каждый месяц предоставляют список запланированных мероприятий. Заведующая садиком планирует из предоставленных списков график мероприятий и информирует родителей о них (цена, условия проведения, дата проведения и т.п.). Родители сообщают заведующей о желании участвовать в каких-либо развлечениях из графика и совершают предоплату за мероприятие. По результатам сообщений родителей формируется окончательный график развлекательных мероприятий.

Сформировать коллекцию данных об участниках мероприятий.

**Задание.**

Определить количество участников каждого мероприятия и его общую стоимость с учетом скидок.

6. **Продажа железнодорожных билетов**. Продажа железнодорожных билетов осуществляется следующим образом – клиент запрашивает наличие билетов на определённую дату и направление, дополнительно указывая тип желаемого вагона (купе, плацкарт и т.д.). Кассир осуществляет поиск в базе по указанным атрибутам, и сообщает клиенту возможные варианты, которые блокируются до окончания сделки. Клиент выбирает из возможных вариантов билет, после чего сообщает кассиру свои паспортные данные для оформления билета. Кассир вносит данные в базу и получает деньги за билет.

Данные, введённые кассиром попадают в базу данных в центральном аппарате, что исключает возможность продажи одного и того же билета разным людям.

Также возможны варианты сдачи билета – в этом случае кассир возвращает клиенту деньги (за вычетом некоторой пени), и забирает оформленный билет, после чего помечает данный билет как незанятый. Пеня на билеты, сданные за 1 месяц до даты использования, составляет 1% от стоимости; сданные за 15 дней – 5%, сданные за 3 дня – 10%, сданные в день использования – 30%.

Сформировать коллекцию данных с информацией о клиентах и проданных им и сданных ими билетах.

**Задание.**

Отобразить информацию о проданных и сданных билетах по каждому направлению.

7. **Работа таксопарка**. Имеется таксопарк, в котором есть штат водителей с автомобилями. Диспетчер автопарка принимает телефонные заказы, фиксируя время заказа, адрес и телефон вызова, а также пункт назначения.

После регистрации заказа диспетчер проверяет возможность его выполнения. Если в таксопарке имеются свободные водители, то они назначаются на выполнение заказа, причём выбирается тот водитель, который ближе всего к клиенту. Если все автомобили заняты, то диспетчер связывается с водителями на маршруте, и выясняет, смогут ли они выполнить заказ в установленное время. После этого диспетчер связывается с клиентом, сообщая ему о возможности/невозможности выполнения заказа.

При выполнении заказа водитель оформляет две квитанции на оплату услуг по перевозке. Клиент расплачивается и расписывается в квитанциях. Одна квитанция остаётся у клиента, а вторую водитель передаёт диспетчеру вместе с деньгами при возвращении в таксопарк.

Сформировать коллекцию данных с информацией обо всех рейсах, осуществленных за день.

**Задание.**

Отобразить информацию по сумме выручки каждого водителя.

8. **Работа склада**. На складе хранится продукция, для которой указывается название, количество, дата поступления и цена. Продукция поступает на склад от поставщиков согласно заключённым договорам.

При поставке товаров бухгалтерия склада посылает запрос поставщику; поставщик высылает счёт-фактуру на оплату заказа. Бухгалтерия осуществляет оплату, и в течении трёх дней продукция поступает на склад.

Для выдачи продукции ведётся учёт заявок от клиентов: клиент посылает запрос на склад, бухгалтерия готовит счёт-фактуру на оплату услуги и передаёт его клиенту. После оплаты счёта заказ передаётся клиенту.

Сформировать коллекцию данных с информацией о проделанных финансовых операциях для руководства склада.

**Задание.**

Отобразить информацию о состоянии заявок по каждому клиенту: количество заказов клиента, число выполненных заказов, число заказов на стадии перевода денег.

9. **Работа Internet магазина**. Системный администратор торговой фирмы размещает на web-сайтах данные о товарах (название, описание, цена). Заказчик осуществляет поиск товара по заданным параметрам. Если поиск удачен, то производится оформление заказа на поставку товара, причём дата доставки указывается заказчиком.

Оплата товара осуществляется наличными, для чего торговая фирма отправляет товар вместе с платёжными документами.

Возникают ситуации, когда товар, описание которого размещено на web-сайте, отсутствует на складе. В этом случае клиент может переоформить заказ или дату поставки.

Системный администратор занимается учётом покупателей, определяя процент скидки постоянным клиентам или клиентам, единовременно оформившим заказ на большую сумму.

Сформировать коллекцию данных с информацией о покупках товаров клиентами.

**Задание.**

Отобразить информацию о покупателях, имя покупателя, общую стоимость приобретений и текущий процент скидки.

10. **Детский садик**. Для вычисления платы за пребывание детей в детском саду утром воспитателями производится учёт поступающих детей. Собранные сведения о пришедших детях передаются заведующей садиком; на основании этих сведений формируется квитанция об оплате (за месяц), которая передаётся родителям через воспитателей. После оплаты квитанция приносится воспитателям. Если в течении 10 дней квитанция не оплачивается, то ребёнок в детский садик не принимается.

Сведения об отсутствующих детях передаются медицинской сестре. Те дети, которые отсутствовали более трёх дней, принимаются в садик только при наличии медицинской справки о здоровье.

Сформировать коллекцию данных с информацией о посещаемости детьми детского сада.

**Задание.**

Отобразить информацию о детях, которым необходимо принести медицинскую справку.

11. **Прокат автомобилей**. Фирма занимается прокатом автомобилей. Имеется парк автомобилей, которые поставляются партнёрами фирмы. Клиенты фирмы обращаются с запросом на получения автомобиля по некоторым параметрам (тип автомобиля, марка, цена и т.д.). Фирма составляет список автомобилей, удовлетворяющих запросу, и высылает его клиенту, после чего осуществляется оформление контракта на прокат автомобиля. При оформлении контракта определяется сумма залога за автомобиль в зависимости от категории риска, в которую попадает клиент. Категория риска зависит от стажа вождения клиента, числа аварий, времени сотрудничества с фирмой и т.д. Для постоянных клиентов руководство фирмы устанавливает персональные скидки.

Руководство фирмы корректирует информацию о поставщиках автомобилей, устанавливает персональные скидки постоянным клиентам, а также получает отчёты о доходах от аренды автомобилей.

Сформировать коллекцию данных с информацией об аренде автомобилей клиентами.

**Задание.**

Отобразить информацию о сумме выручки по каждому типу автомобиля с учетом скидок.

12. **Покупка квартир.** В брокерскую фирму обращаются клиенты, которые хотят купить квартиру. Брокер регистрирует параметры квартиры (метраж, расположение, максимально допустимую цену и т.д.), после чего осуществляет поиск подходящих предложений в базе данных фирмы. Список найденных квартир предоставляется клиенту, который выбирает из списка заинтересовавшие его предложения, после чего информация о соответствующих квартирах блокируется. Квартиры, помеченные как «заблокированные» не рассматриваются при поиске.

Клиент и брокер производят осмотр квартир из списка, после чего клиент выбирает наиболее подходящую квартиру, которая переводится в статус «сделка». Статус «заблокирована» с остальных квартир из списка снимается.

Руководство брокерской фирмы занимается:

* наполнением базы данных о продаваемых квартирах из некоторых внешних (газеты, объявления) и внутренних источников;
* оформлением договоров о купле/продажи квартир со статусом «сделка»;

Сформировать коллекцию данных с информацией о работе брокеров.

**Задание.**

Отобразить информацию о сделках по каждому брокеру.

13. **Работа ресторана**. Клиент приходит в ресторан и сдаёт верхнюю одежду в гардероб. Если столик зарезервирован, или есть свободные места, то метрдотель размещает клиента за столиком и вызывает официанта. Официант приносит меню и записывает заказ на бланке, который передаётся шеф-повару. После выполнения заказ приносится клиенту. После обслуживания клиент расплачивается, получая чек об оплате.

Меню ресторана составляется шеф-поваром исходя из имеющихся продуктов. При необходимости шеф-повар оформляет заказ на продукты, который передаётся администратору ресторана. Для выполнения заказа администратор связывается с поставщиками провизии, оплачивает заказ, и указывает день поставки.

В конце рабочего дня администратор:

* подсчитывает выручку, полученную за день;
* формирует список зарезервированных столиков на следующий день;
* получает меню на следующий день от шеф-повара.

Сформировать коллекцию данных с информацией о заказах за день.

**Задание.**

Отобразить информацию о выручке по каждому блюду: название блюда, число заказов и суммарная выручка.

14. **Продажа квартир**. В брокерскую фирму обращаются клиенты, желающие продать квартиру. Фирма определяет брокера, который будет вести дела клиентов, на основании текущей загруженности работников. Выделенный брокер регистрирует параметры продаваемой квартиры и контактные адреса клиента и вносит их в центральную базу фирмы.

При возникновении запроса на квартиру брокер связывается с клиентом и согласовывает удобное время и дату осмотра квартиры. В случае нахождения покупателя на квартиру брокер связывается с клиентом, согласовывает дату оформления договора о купле/продаже. Для осуществления сделки брокер оформляет необходимые свидетельства, контракты и договора, после чего передаёт их центральный аппарат фирмы для конечного нотариального заверения.

Сформировать коллекцию данных с информацией о клиентах брокерской фирмы с приставленным брокером и статусом продажи.

**Задание.**

Отобразить информацию характеризующую работу каждого из брокеров (общее число дел, число завершённых сделок).

15. **Стоматологическая клиника**. Пациент обращается в стоматологическую клинику с жалобой. Консультант опрашивает пациента, фиксирует его жалобы, проводит осмотр, после чего назначает необходимые процедуры, лечение и стоимость.

После консультации пациент направляется к одному из лечащих врачей (исходя из заболеваний пациента), где проходит необходимый курс лечения.

Сформировать коллекцию данных с информацией о пациентах их заболеваниях и лечащих врачах.

**Задание.**

Отобразить информацию по количеству пациентов каждого лечащего врача и прибыли, которую врач принес клинике.

16. **Работа терапевта**. Пациент обращается к терапевту с жалобой на плохое самочувствие. Врач опрашивает пациента, фиксирует его жалобы, проводит осмотр, после чего назначает обследование и формирует направления на анализы. Пациент обращается в диагностический центр, где проходит обследование и сдаёт анализы.

Результаты обследования и анализов из диагностического центра поступают врачу, который ставит диагноз, назначает курс лечения и выписывает рецепты на медикаменты. При необходимости терапевт привлекает врачей-специалистов для уточнения диагноза.

Пациент приобретает медикаменты, принимает их и проходит лечебные процедуры. После окончания лечения пациент приходит к лечащему врачу для проведения контроля самочувствия.

Каждый месяц врач-терапевт формирует отчёт по заболеваниям на его участке, который передаёт главврачу.

Сформировать коллекцию данных с информацией по заболеваниям пациентов за 1 месяц.

**Задание.**

Отобразить информацию отражающую заболеваемость за 1 месяц: диагноз, число заболевших пациентов.

17. **Работа почты**. На почту приходят клиенты, желающие отправить посылку, заказное письмо или бандероль. В зависимости от вида отправления указывается адрес отправителя и получателя, категория посылки, вес, размер и тип отправки (первым классом, второй класс). Дополнительно клиент может застраховать отправление.

Работник почты оформляет все реквизиты и формирует квитанцию об оплате, корешок которой выдаётся клиенту. После этого посылка относится в отдел отправки почты, причём для каждого вида отправлений предусмотрены отдельные стеллажи.

В конце рабочего дня оформляется общий отчёт по стеллажам, в которых указывается тип отправления, пункт назначения, вес и размер. Данный отчёт передаётся перевозчикам почты вместе со всеми отправлениями.

Если почта не доходит до получателя, она отправляется назад к отправителю. Если посылка приходит в плохом состоянии получатель может воспользоваться страховкой, если посылка была застрахованная либо вернуть посылку назад отправителю. После успешного получения посылки, отправитель получает уведомление об успешной доставке.

Сформировать коллекцию данных для создания отчёта о работе почты за месяц.

**Задание.**

Отобразить информацию по каждому виду услуг (количество доставленных, забракованных и возвращенных писем посылок и бандеролей).

18. **Работа аэропорта**. В аэропорту существует два терминала: терминал А для внутренних рейсов и терминал Б – для между народных. Пассажир прибывает в аэропорт и становится на регистрацию к соответствующему терминалу. При регистрации проверяются документы и авиабилет пассажира, при международном перелёте дополнительно проверяется виза пассажира. Во время регистрации дополнительно проверяется, не состоит ли пассажир в розыске милиции или Интерпола, при необходимости информируются соответствующие органы.

В случае удачной регистрации происходит оформление и проверка багажа. Если объём или вес багажа пассажира превышает допустимые нормы, то выписывается квитанция на оплату перевозки.

За 10 мин до отправки рейса регистрация пассажиров прекращается, формируется окончательный список пассажиров и багажа. На основании сформированных списков владельцы авиакомпаний принимают решение о сокращении или увеличении числа рейсов:

* если количество пассажиров меньше заданной нормы рентабельности рейса, то принимается решение по сокращению количества рейсов.
* если самолет загружен на 95% то рассматривается вариант по увеличению частоты полетов рейсов в данном направлении.

Сформировать коллекцию данных для создания списка пассажиров и багажа.

**Задание.**

Сформировать представление для принятия решения о рентабельности рейса.

19. **Работа СТО**. СТО оказывает следующий ряд услуг: починка транспортного средства (ТС) и диагностика (с оценкой стоимости).

Если клиент приезжает для починки своего ТС, то производится диагностика для определения причины неисправности Т, при этом:

* записываются данные о машине, владельце и причины поломки,
* определяется комплект деталей, которые нужно заменить.

На основании этих данных клиенту сообщают срок, в течении которого будет сделан ремонт ТС, и предварительная сумма ремонта.

Для ремонта автомобиля СТО использует собственные детали; если необходимый комплект деталей отсутствует, то мастер оформляет заказ комплекта на складе.

Если ремонт был произведен раньше назначенного срока, то владельцу сообщают о возможности забрать ТС. После окончания ремонта СТО оформляет гарантию на произведённый ремонт; если в течении гарантийного срока возникает эта же поломка, то её ремонт производится бесплатно.

Если клиент желает произвести диагностику ТС с оценкой стоимости, то сначала определяется вид диагностики (полная или поверхностная). После этого ТС устанавливается в специальную диагностическую машину, которая выдает полный отчет о текущем состоянии ТС. На основании отчёта производится оценка стоимости автомобиля.

Сформировать коллекцию данных о выполненных СТО работах.

**Задание.**

Отобразить информацию о том, каких и сколько работ было выполнено.

20. **Работа спорткомплекса**. Спорткомплекс предлагает следующий набор услуг: бассейн, фитнесс центр, тренажерный зал, массаж и сауна. Каждый клиент спортзала получает клубную карточку, при оформлении которой учитывается набор услуг, которыми пользуется клиент. Также учитывается режим посещения спортзала:

* неограниченный – клиент посещает спорткомплекс на протяжении определенного срока, без ограничения в количестве посещений;
* месячный абонент – посещение спорткомплекса в установленное время на протяжении месяца (деньги за пропущенные занятия не возмещаются);
* разовый – оплата по каждому занятию отдельно.

При выборе бассейна клиент должен дополнительно пройти медицинское обследование для получения справки о состоянии здоровья. Обследование производится в медицинской клинике, являющейся партнёром спорткомплекса, результаты обследования автоматически направляются в спорткомплекс.

На основании услуг, заказанных клиентами спортзала, администрация спорткомплекса формирует детальный отчёт о наиболее популярных услугах. На основании отчёта принимается решение о возможном расширении перечня услуг, об улучшении отдельных видов услуг и т.д.

Сформировать коллекцию данных о клиентах спорткомплекса и услугах которыми они пользуются.

**Задание.**

Отобразить виды услуг и их количество.

21. **Работа паспортного стола**. Паспортный стол занимается следующими задачами: выдача гражданского паспорта, выдача заграничного паспорта, регистрация граждан при смене места жительства (выписка и прописка), обновление паспорта.

Для получения паспорта гражданин обязан предоставить следующий набор документов: свидетельство о рождении, справка с ЖЭКа (об оплате коммунальных услуг), фотография 3х4 и т.д.. Если гражданину более 18 лет, и он получает паспорт впервые, то кроме документов необходимо предоставить квитанцию об уплате штрафа за просрочку.

При переоформлении паспорта в случае утраты гражданин пишет заявление о выдаче нового паспорта, в котором указывается причина переоформления штрафа. Если паспорт утерян, то гражданин обязан оплатить штраф. Если паспорт украден, то паспортный стол выполняет запрос в милицию о наличии возбуждённого дела о краже. Ответ из милиции поступает в двухнедельный срок; при подтверждении информации паспорт гражданину выдают бесплатно.

Оплата штрафов осуществляется в филиале банка, расположенного в задании паспортного стола. Установкой процента банка за услуги определяется директором паспортного стола.

При получении загранпаспорта гражданин обязан предоставить гражданский паспорт, справку об отсутствии непогашенных судимостей, справку из военкомата (для находящихся на военном учёте).

Для оформления смены места жительства, гражданин должен принести:

* при выписке – справку об оплате коммунальных услуг с места текущей прописки и заявление на выписку, паспорт.
* при прописке – паспорт с отметкой о выписке с предыдущего места жительства, заявление на прописку, справки о возможности прописки.

Сформировать коллекцию данных с отчетом о работе паспортного стола.

**Задание.**

Отобразить виды услуг и их количество.

22. **Работа интернет аукциона**. В аукционе принимают участие покупатели и продавцы. Продавцы выставляют товар на аукцион и выбирают тип продажи товара: продать сразу, выставить на аукцион, договорной.

При определении типа «продать сразу» продавец указывает стоимость товара, срок, в течении которого товара выставляется на аукционе, описание товара и условия доставки (стоимость, срок и тип). При определении типа «выставить на аукцион» продавец определяет начальную ставку, срок, описание товара и условия по перевозке. При выборе типа «договорной» продавец указывает только начальную стоимость и описание товара, оставляя за работниками аукциона право определять срок выставки товара и услуги по доставке.

Покупатель просматривает товары, выставленные на аукционе, по категориям. Если он нашёл заинтересовавшее его предложение, то он регистрируется, после чего либо приобретает товар, либо участвует в аукционе.

Если товар выставлен для продажи на аукционе, то для приобретения его покупатель должен «перебить» ставку до окончания срока выставления товара. При окончании срока выставления товара работники аукциона определяют, кто из покупателей предложил наибольшую ставку, и уведомляют его о возможности получить товар.

При оплате товара покупателю предлагается указать вид платежа: WEB-money, оплата со счёта, или другие виды. После произведения оплаты работники аукциона переводят деньги на счёт продавца (за вычетом комиссии), и сообщают ему контактную информацию о клиенте для доставки товара.

Сформировать коллекцию данных с отчетом о работе аукциона.

**Задание.**

Отобразить данные по покупателям: имя покупателя, тип покупки и число покупок указанного типа.

23. **Работа кадрового агентства**. Кадровое агентство занимается подбором персонала для предприятий и поиском вакансий для людей.

При осуществлении подбора персонала предприятие обращается в кадровое агентство и оформляет заявку на поиск работника. В заявке указывается название вакансии, профессиональные требования к претенденту, предлагаемый уровень заработной платы, наличие социального пакета и п. Работники агентства размещают заявление на сайте агентства, подают объявление в соответствующие печатные издания, указывая в качестве контактов свои рабочий телефон и e-mail. Претенденты на вакансию обращаются в кадровое агентство, где проходят начальное собеседование. Если работник кадрового агентства считает, что претендент удовлетворяет выставленным требованиям, то он записывает его данные (ФИО, контактный телефон или e-mail) и сообщает их предприятию.

При осуществлении подбора вакансий клиент заполняет бланк, в котором указывает свои умения, паспортные данные, желаемый уровень заработной платы. Работник агентства производит поиск вакансий по указанным параметрам, при нахождении вакансии клиенту сообщаются контактные данные работодателя.

Сформировать коллекцию данных о клиентах кадрового агентства, которые благодаря ему нашли работу.

**Задание.**

Определить самые успешные для агентства предприятия.

24. **Поселение студентов 1го курса**. По результатам вступительных экзаменов каждый факультет института формирует список студентов 1-го курса, нуждающихся в выделении места в общежитии; списки передаются коменданту общежитий.

В это же время комендант запрашивает о числе свободных мест каждое из общежитий института. Полученная информация от общежитий сводится в общий отчёт.

Распределение студентов осуществляется следующим образом: сначала выбираются те студенты, которые имеют льготы (сироты, чернобыльцы, инвалиды и т.д.).

Этих студентов пытаются распределить в общежитие, соответствующее их факультету. Если это не удаётся, то им выдают ордер на поселение в произвольное общежитие.

Далее распределяются остальные студенты по общежитиям соответствующих факультетов. Если студент из данной категории не получает место в общежитии своего факультета, то он переносится в общий список желающих получить место. Студенты из этого списка расселяются по свободным местам в порядке их возникновения.

Сформировать коллекцию данных для получения информации о заселении общежитий: общее число мест, число свободных мест, число претендентов.

**Задание.**

Определить нехватку или перебор свободных мест.

25. **Размещение рекламы в журнале**. В отдел рекламы издательства обращаются клиенты, желающие разместить рекламу. При регистрации заказа приёмщиком фиксируется макет рекламы (текст, оформление, графические элементы и т.д.).

Если макета нет, то клиент может заказать его на фирме, являющейся партнёром издательства. При этом отделом рекламы фиксируются пожелания клиента, которые передаются на фирму. Фирма создаёт несколько макетов рекламы, из которых клиент выбирает наиболее понравившийся.

Принятый заказ передаётся подборщику, который составляет список возможных размещений макета с указанием стоимости, после чего передаёт этот список приёмщику и блокирует всё позиции списка.

Клиент просматривает составленный список, отбирает подходящие варианты расположения и оплачивает их. После этого приёмщик передаёт информацию подборщику о выбранных вариантах, остальные позиции списка помечаются как свободные.

Сформировать коллекцию данных о клиентах и их заказах со стоимостью.

**Задание.**

Определить прибыль по каждому макету.

26. **Работа библиотеки**. В студенческую библиотеку обращаются студенты, желающие получить учебную литературу. При оформлении заявки на литературу необходимо получить бланк заказа, в котором надо указать точное название книги, авторов и год издания; бланк заказа можно получить библиотекаря соответствующего отдела.

Если какие-то реквизиты книги не известны, то студент направляется в каталог библиотеки для заполнения бланка. Заполненный бланк отдаётся в отдел библиотеки, где происходит проверка студента. Если на студенте числится более 10 книг, то в получении новой книги ему отказывают.

Поданные заявки передаются в хранилище библиотеки, где по каждой заявке или находится книга, или указывается на бланке заказа её отсутствие. После этого все рассмотренные заказы передаются в зал выдачи заказов, где и забираются студентами.

Сформировать коллекцию данных о студентах их заказах и статусе выполнения.

**Задание.**

Отобразить информацию о количестве затребованной литературы.

27. **Детский садик**. Для вычисления платы за пребывание детей в детском саду утром воспитателями производится учёт поступающих детей. Собранные сведения о пришедших детях передаются заведующей садиком; на основании этих сведений формируется квитанция об оплате (за месяц), которая передаётся родителям через воспитателей. После оплаты квитанция приносится воспитателям. Если в течении 10 дней квитанция не оплачивается, то ребёнок в детский садик не принимается.

Сведения об отсутствующих детях передаются медицинской сестре. Те дети, которые отсутствовали более трёх дней, принимаются в садик только при наличии медицинской справки о здоровье.

Сформировать коллекцию данных с информацией о посещаемости детьми детского сада.

**Задание.**

Отобразить информацию об оплате за пребывание детей в деском саду, по каждому ребенку.

28. **Работа интернет аукциона**. В аукционе принимают участие покупатели и продавцы. Продавцы выставляют товар на аукцион и выбирают тип продажи товара: продать сразу, выставить на аукцион, договорной.

При определении типа «продать сразу» продавец указывает стоимость товара, срок, в течении которого товара выставляется на аукционе, описание товара и условия доставки (стоимость, срок и тип). При определении типа «выставить на аукцион» продавец определяет начальную ставку, срок, описание товара и условия по перевозке. При выборе типа «договорной» продавец указывает только начальную стоимость и описание товара, оставляя за работниками аукциона право определять срок выставки товара и услуги по доставке.

Покупатель просматривает товары, выставленные на аукционе, по категориям. Если он нашёл заинтересовавшее его предложение, то он регистрируется, после чего либо приобретает товар, либо участвует в аукционе.

Если товар выставлен для продажи на аукционе, то для приобретения его покупатель должен «перебить» ставку до окончания срока выставления товара. При окончании срока выставления товара работники аукциона определяют, кто из покупателей предложил наибольшую ставку, и уведомляют его о возможности получить товар.

При оплате товара покупателю предлагается указать вид платежа: WEB-money, оплата со счёта, или другие виды. После произведения оплаты работники аукциона переводят деньги на счёт продавца (за вычетом комиссии), и сообщают ему контактную информацию о клиенте для доставки товара.

Сформировать коллекцию данных с отчетом о работе аукциона.

**Задание.**

Отобразить данные по продавцам: имя продавца, тип продажи и число продаж указанного типа.

29. **Гостиница**. Гостиница предоставляет номера клиентам на определённый срок. Каждый номер характеризуется вместимостью, комфортностью (президентский, люкс, полулюкс, обычный) и ценой. При поселении в гостиницу клиент указывает требуемые характеристики номера; если имеются подходящие свободные номера, то клиент занимает номер. При этом фиксируются данные клиента (ФИО, контактный телефон, номер паспорта), дата поселения, и определяется вид залога (документы или деньги).

При выезде из гостиницы администратор гостиницы формирует счёт на оплату, который передаётся клиенту. Клиент оплачивает счёт в филиале некоторого банка, являющегося партнёром гостиницы и расположенного на её территории. Оплаченный счёт клиент передаёт администратору, после чего ему возвращается залог.

Сформировать коллекцию данных для получения информации о использовании номеров: вид номера, количество номеров данного вида, число занятий номеров.

**Задание.**

Отобразить выручку по каждому типу номера.

30. **Работа СТО**. СТО оказывает следующий ряд услуг: диагностика транспортного средства (ТС) для страховой компании и тюнинг ТС.

Если клиент производит диагностику для страховой компании с оценкой ущерба при аварии, то определяются те части ТС, которые были повреждены; после чего по каталогу оценивается стоимость ущерба, и формируется акт о стоимости повреждений.

Если клиент желает выполнить тюнинг транспортного средства, то сначала определяется вид тюнинга: внешний, внутренний, арт-тюнинг или смешанный. При выборе внешнего тюнинга ТС направляется в дизайнерскую студию, являющуюся партнёром СТО. Там снимается цифровая фотография ТС, на основании снимков проектируется ряд возможных изменений (при помощи добавления спойлеров, подвесов и т.д.). Клиент выбирает понравившиеся проекты, на основании которых СТО выполняет тюнинг.

При выборе арт-тюнинга клиент высказывает свои пожелания мастеру-художнику, который формирует ряд эскизов для нанесения на ТС. Клиент выбирает понравившийся, и эскиз наносится на ТС.

При выборе внутреннего тюнинга клиент определяет перечень изменений, которые надо произвести в салоне: тонировка, сигнализация, центральный замок и т.д. (необходимая фурнитура заказывается со склада).

При выборе смешанного тюнинга выполняются все три вида тюнинга.

При любом виде тюнинга СТО:

* фиксирует данные о клиенте (ФИО, контактный телефон) и ТС,
* формирует платёжный документ с указанием стоимости услуги.

Сформировать коллекцию данных для получения информации о клиентах и типе услуг, которые были выполнены.

**Задание.**

Отобразить выручку по каждому типу тюнинга.

# Требования к отчету

Отчет подается после полной сдачи и защиты лабораторной работы в электронном виде (документ Word).

Отчет должен быть оформлен согласно ДСТУ 3008-95.

В отчет должен содержать следующие пункты:

* титульный лист;
* содержание;
* цель работы;
* постановка задачи;
* аналитические выкладки;
* пошаговый алгоритм решения расчётной части задачи и подзадач (при необходимости);
* UML-диаграмму классов;
* исходный код программы с комментариями;
* примеры работы программы;
* выводы, с обоснованием результата.

**Отчет оценивается максимально в 0,5 балла.**

# Контрольные вопросы