Пермский Национальный Исследовательский  
Политехнический Университет

**Лабораторная работа № 6**

Информатика

за 2 семестр

Вариант № 1

Выполнил:

Студент группы РИС 20-1-бз

КургановН.В.

20-ЭТФ-631

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

2021

**АТД. Контейнеры**

1. **Цель задания**.
   1. Создание консольного приложения, состоящего из нескольких файлов в системепрограммированияVisual Studio.
   2. Реализациякласса-контейнера.
2. **Задание.**

Класс-контейнер ВЕКТОР с элементами тип аint. Реализовать операции:

[] – доступа по индексу;

() – определение размера вектора;

+ число–добавляет константу ко всем элементам вектора;

++- переход к следующему элементу (с помощью класса-итератора).

1. **Файлmain.cpp**

#include"Vector.h"

#include<iostream>

usingnamespacestd;

voidmain()

{

Vectora(5);

cout<< a<<"\n";

cin>> a;

cout<< a<<"\n";

a[2] = 100;

cout<< a<<"\n";

Vectorb(10);

cout<< b<<"\n";

b = a;

cout<< b<<"\n";

Vectorc(10);

c = b + 100;

cout<< c<<"\n";

cout<<"\nthe length of a="<<a()<<endl;

cout<<\*(a.first())<<endl;

Iterator i = a.first();

++i;

cout<<\*i <<endl;

for(i=a.first(); i!=a.last(); ++i) cout<<\*i<<endl;

for (i=a.first(); i!=a.last(); i++) cout<<\*i<<endl;

}

1. **ФайлVector.cpp**

#include"Vector.h"

Vector::Vector(ints, intk)

{

size = s;

data = newint[size];

for (inti = 0; i< size; i++)

data[i] = k;

beg.elem = &data[0];

end.elem = &data[size];

}

Vector::Vector(constVector&a)

{

size = a.size;

data = newint[size];

for (inti = 0; i< size; i++)

data[i] = a.data[i];

beg=a.beg;

end=a.end;

}

Vector::~Vector()

{

delete[]data;

data = 0;

}

int&Vector::operator[](intindex)

{

if (index< size) return data[index];

elsecout<<"\nError! Index>size";

}

VectorVector::operator+(constintk)

{

Vectortemp(size);

for (inti = 0; i< size; ++i)

temp.data[i] += data[i] + k;

returntemp;

}

intVector::operator()()

{

returnsize;

}

Vector&Vector::operator=(constVector&a)

{

if (&a == this) return \*this;

size = a.size;

if (data != 0) delete[]data;

data = newint[size];

for (inti = 0; i< size; i++)

data[i] = a.data[i];

beg=a.beg;

end=a.end;

return \*this;

}

istream&operator>>(istream&in, Vector&a)

{

for (inti = 0; i<a(); i++)

in>>a.data[i];

returnin;

}

ostream&operator<<(ostream&out, constVector&a)

{

for (inti = 0; i<a.size; i++)

out<<a.data[i] <<" ";

returnout;

}

1. **ФайлVector.h**

#pragmaonce

#include<iostream>

usingnamespacestd;

classIterator

{

friendclassVector;

public:

Iterator() { elem = 0; }

Iterator(constIterator&it) { elem = it.elem; }

booloperator == (constIterator&it) { returnelem == it.elem; }

booloperator != (constIterator&it) { returnelem != it.elem; }

voidoperator++ () { ++elem; }

voidoperator ++(int) { elem++; }

voidoperator --() { --elem; }

int&operator \*() const{ return \*elem; }

private:

int\* elem;

};

classVector

{

public:

//конструкторы с параметрами

//s - длина, k - значение каждого элемента

Vector(ints, intk = 0);

//клонирование

Vector(constVector&a);

//деструктор

~Vector();

//перегрузка присваивания

Vector&operator= (constVector&a);

//перегрузка обращения по индексу

int&operator[] (intindex);

//перегрузка добавления

Vectoroperator+(constintk);

//возвращениедлинывектора

intoperator()();

Iteratorfirst() { return beg; };

Iteratorlast() { return end; }

//перегрузкавводавывода

friendistream&operator>>(istream&in, Vector&a);

friendostream&operator<<(ostream&out, constVector&a);

private:

int size;

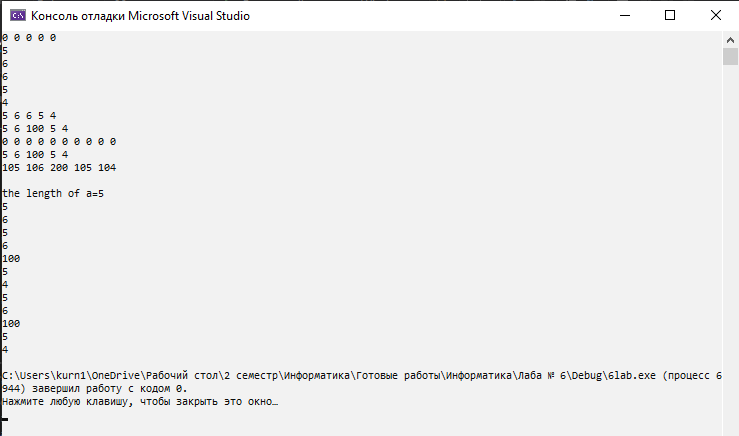
int\* data;

Iteratorbeg;

Iteratorend;

};

1. **Результат работы программы**

****

1. **Контрольные вопросы.** 
   1. Чтотакоеабстрактныйтипданных? ПривестипримерыАТД.

Тип данных, определяемы только через операции, которые могут выполняться над соответствующими объектами например атд можно считать использование функции для разных классов которые предком которых является 1 и тот же абстрактный класс.

* 1. Привестипримерыабстракциичерезпараметризацию.

Использование 1 и той-же функции для разных переменных

* 1. Привестипримерыабстракциичерезспецификацию.

Использование 1 и той-же функции для разных классов из 1

* 1. Чтотакоеконтейнер?Привестипримеры.

Контейнер - набор однотипных элементов. Частный случай массив.

* 1. Какиегруппыоперацийвыделяютвконтейнерах?

Операции доступа к элементам, которые обеспечивают и операцию замены значений элементов

Операции добавления удаления

Операция поиска

Операция объединения контейнеров

Специальные операции, зависящие от вида контейнера

* 1. Какиевидыдоступакэлементамконтейнерасуществуют?Привестипримеры.

Прямой доступ - доступ по индексу (a[10] - 10 элемент массива a)

Ассоциативный доступ - доступ по идентификатору который выступает в роли индекса (a["abc"] передаёт элемент с именем abc )

Последовательный доступ - от элемента к элементу при помощи различный функций (a.first() - передаёт первый элемент)

* 1. Чтотакоеитератор?

Итератор это объект, который обеспечивает последовательный доступ к элементам контейнера.

* 1. Какимобразомможетбытьреализованитератор?

Итератор можно реализовать как класс, представляющий такой же набор операций. Итератор реализуется как класс, который имеет такой же интерфейс.

* 1. Какимобразомможноорганизоватьобъединениеконтейнеров?

Можно реализовать как объединения множеств в новый контейнер попадают только те элементы которые есть хотя бы в 1 контейнере.

* 1. Какойдоступкэлементампредоставляетконтейнер,состоящийизэлементов

«ключ-значение»?

Предоставляет доступ к значению элемента по ключу (ассоциативный контейнер)

* 1. Как называется контейнер, в котором вставка и удаление элементов выполняетсянаодном конце контейнера?

Lastinfirstout

* 1. Какойизобъектов (a,b,c,d) является контейнером?

D

* 1. Какой из объектов (a,b,c,d)не является контейнером?

C

* 1. Контейнер реализован как динамический массив, в нем определена операциядоступпоиндексу.Какимбудетдоступкэлементамконтейнера?

Прямой доступ

* 1. Контейнерреализованкаклинейныйсписок.Какимбудетдоступкэлементамконтейнера?

Последовательный доступ