I. Задан интерфейс класса (в соответствие с вариантом задания).

II. Выполнить:

1. Дополнить класс недостающими закрытыми свойствами и методами.
2. Перегрузить конструктор класса.
3. Один(или несколько) методов реализовать как встроенный.
4. Выполнить реализацию каждого метода, указанного в описании класса.
5. Описание класса выполнить в отдельном заголовочном файле (.h), реализации класса – в отдельном исходном файле (.cpp).
6. Собрать проект, в main() реализовать меню, позволяющее:

\* создавать и удалять объект класса

\* производить предусмотренные вариантом задания операции с объектом

\* выводить на экран информацию об объекте (состояние).

Вариант 15: Класс упорядоченная последовательность целых чисел (числа могут повторяться).

class SortSeq

{

public:

SortSeq(); // пустая последовательность

~SortSeq();

void add(int x); // добавить число

int count();

int &get(int i); // получить элемент последовательности

void remove(int i); // удалить i-й элемент

void print(); // распечатать состояние объекта

};

**Листинг описания класса:**

class SortSeq

{

private:

vector<int> seque;

public:

SortSeq(); //Empty sequence конструктор

SortSeq(vector <int> data) //Заполненная последовательность случайными числами

{

seque = data;

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

}

~SortSeq(); //деструктор

void add(int x) //добавление числа – встроенный метод

{

seque.push\_back(x);

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

}

int count(); //Получения количества элементов в последовательности

int&get(int i); //Получение элемента последовательности по индексу

void remove(int i); //Удаление элемента последовательности по индексу

void print(); //Вывод последовательности

};

**Листинг реализации класса**

SortSeq::SortSeq() //Конструктор

{

}

SortSeq::~SortSeq() //Конструктор

{

}

int SortSeq::count() // Получение количества элементов

{

if(seque.empty())

{

return 0;

}

else

{

return seque.size();

}

}

int &SortSeq::get(int i) //Получение элементов

{

return seque.at(i);

}

void SortSeq::remove(int i) //Удаление элементов

{

seque.erase(seque.begin() + i);

}

void SortSeq::print() //Печать элементов

{

cout<<"{";

for(int pi=0; pi<seque.size(); pi++)

{

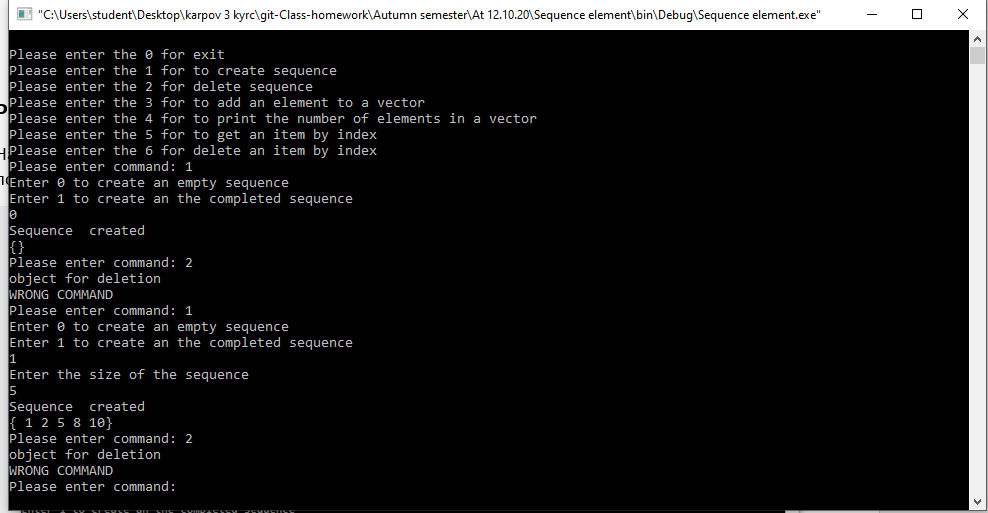
cout<<" "<<seque.at(pi);

}

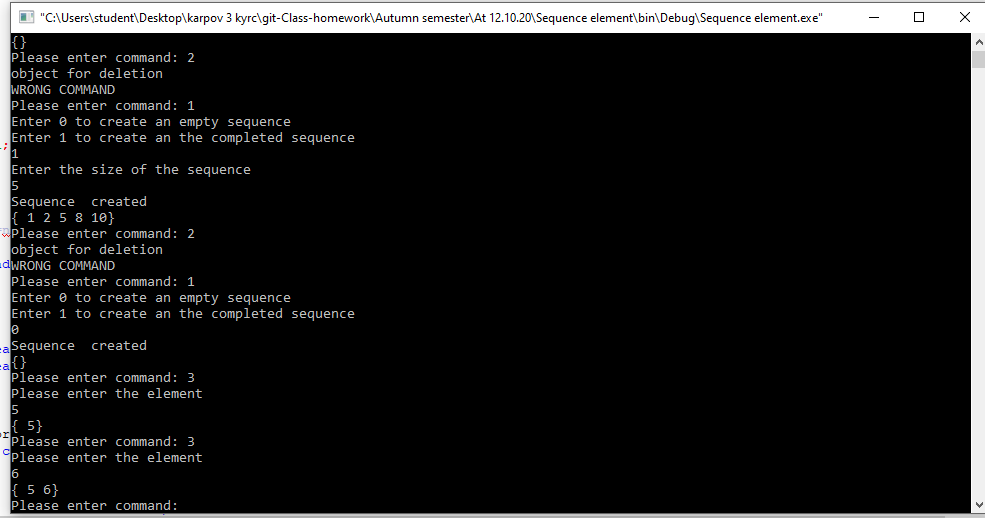
cout<<"}"<<endl;}

**Результаты работы программы проверки класса**

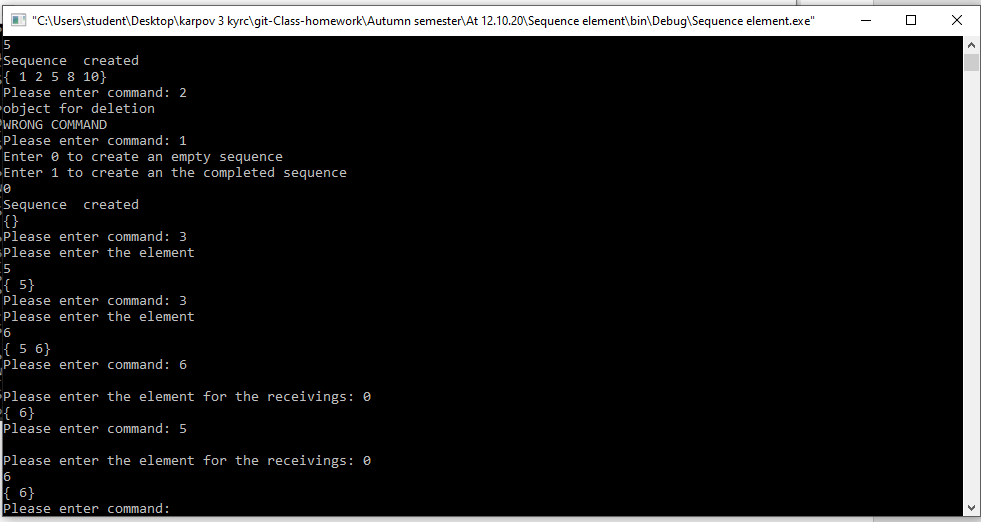
На данном скриншоте я проверял работу программы при создании и удалении последовательности.



Затем приступил к добавлению элементов в последовательность.



Далее некоторые элементы удалял, некоторые выводил на экран по индексу.



После проверил программу на возможные сбои и добавил некоторый элемент в последовательность, прекратил работу программы с помощью соответствующего пункта меню.

