Билет №14

Задание 1

Познакомьтесь с предложенной реализацией классов. Определите тип отношения между классами (предоставьте описание в виде UML), ответ обоснуйте.

Объявление класса ClassA

```
#include "ClassB.h"
class ClassB;
class ClassA
{
	int m_n; //фактическое число частей
	ClassB* m_pB[100]; //массив указателей на части
public:
	 ClassA();
	~ClassA();
	const int getN () const; //вернуть число частей
	const ClassB** getB (int&) const; //вернуть части
	bool add (const int); //добавить часть
	bool del (const int); //удалить часть
	int find (const int) const; //определить индекс части
	const int getX(const int) const;//получить значение объекта
};
```

Реализация класса ClassA

```
#include "ClassA.h"
ClassA::ClassA(): m_n (0) {}
ClassA :: ~ClassA()
{
	for (int j(0); j < m_n; j++)
		delete m_pB[j];
}
//вернуть количество частей
const int ClassA:: getN() const {return m_n;}
//вернуть части
const ClassB** ClassA:: getB (int& n) const
{
	n = m_n;
	return const_cast <const ClassB**> (m_pB);
}
//добавить часть
//метод получает все необходимые параметры для создания части
bool ClassA:: add (const int x)
{
	if (find(x) >= 0) return false;
	m_pB[m_n] = new ClassB;
	m_pB[m_n]->setX(x);
	m_n++;
	return true;
}
//удалить часть
bool ClassA:: del(int key)
{
	int jDel = find (key);
	//поиск индекса части
	if (jDel < 0) return false;
	//часть не найдена
	delete m_pB[jDel];
	//разрушение части
	while (jDel < m_n - 1)
	{
		//сжатие массива
		m_pB[jDel] = m_pB[jDel+1]; jDel++;
	}
	m_n--;
```

```
	return true;
}
//вернуть индекс части
int ClassA :: find (const int key) const
{
	for (int j(0); j < m_n; j++)
		if (m_pB[j]->verify(key))
			return j;
	return -1;
}
//вернуть значение объекта с индексом
//взывается метод класса частей m_pB[j]->getX();
const int ClassA:: getX (const int j) const
{
	return m_pB[j]->getX();
}
```

Объявление класса ClassB

```
class ClassB
{
	int m_x; //член-данное части
public: ClassB();
		~ClassB();
//методы установки и получения значения m_x
		void setX(const int);
		const int getX() const;
		bool verify (const int) const; //проверить объект
		ClassB& operator = (const ClassB&);//перегрузка оператора =
		//перегрузка операторов == и !=
		bool operator == (const ClassB&) const;
		bool operator != (const ClassB&) const;
};
```

Реализация класса ClassB

```
#include "ClassB.h"
ClassB::ClassB(): m_x(0) {}
ClassB :: ~ClassB(){} //установить m_x
void ClassB:: setX(const int x){m_x = x;}
//получить значение т_х
const int ClassB:: getX() const {return m_x;}
//перегрузка оператора =
ClassB& ClassB ::operator = (const ClassB& r)
{
	if (this != &r) m_x = r.m_x; return *this;
}
//перегрузка оператора ==
bool ClassB :: operator == (const ClassB& r) const {return m_x == r.m_x;}
//перегрузка оператор !=
bool ClassB :: operator != (const ClassB& r) const {return m_x != r.m_x;}
```

Задание 2

Базовые принципы проектирования (SOLID). Принцип – единственной ответственности. На представленном примере определить, существует ли нарушение принципа единственной ответственности. Если да, **то предложите решение**. Ответ обоснуйте.

Легенда

Рассмотрим следующий фрагмент псевдокода. Здесь представлен класс, обеспечивающий перенос данных с сервера на другой сервер.

```
class DataMigrater
{
public:

IList <serverdata> GetData(string initialDate, string endDate)
{
	// получить данные сервера в пределах диапазона дат
	return NewList<serverdata>();
	}

	 IList<serverdata>ProcessData(IList <serverdata >&rawData)
	{
		//применяем правила по обработке данных.
		return rawData.ToList();
	}

	void Migrate(IList <serverdata >&rawData)
	{
	 // перенос обрабатываемых данных с сервера на сервер
	}
}
```

Задание 3

Паттерн «Шаблонный метод». Рассмотреть назначение и архитектуру ***шаблонного метода***. Применить к предложенной легенде рассматриваемый паттерн. Решение представить в виде кода с подробными объяснениями в онлайн компиляторе, также предоставить UML диаграмму для предложенного решения.

Легенда

Необходимо реализовать класс класса под названием **SecurityManager**, в котором присутствует метод по созданию пользователя в системе (void CreateUser(string username, string realName, string password). При создании пользователя выполняются следующие действия: шифруется пароль пользователя, вставляются все необходимые записи в базу данных, а также, по желанию, заполняется журнал аудита о создании пользователя.

Требования к ответам

Задание представляете в файле типа word. Каждое задание снабжаете подробными комментариями (обоснованиями предложенного решения). UML диаграмму отдельной картинкой (можно от руки нарисовать), также с объяснениями + ссылка на код в онлайн компиляторе. https://www.onlinegdb.com/

Ответ, скопированный откуда-либо в качестве ответа на задание, не рассматривается, такая работа не зачитывается.