Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.2**

Дисциплина: основы алгоритмизации и программирования

Тема: “Классы и объекты. Использование конструкторов.”

Вариант 10

Выполнил:

студент группы ИВТ-20-2Б Галинов О.Ю.

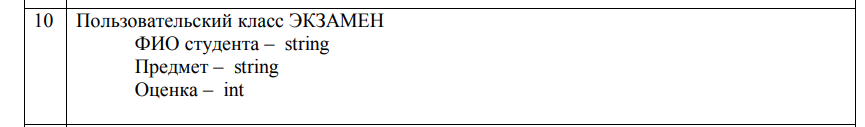
Проверила: доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Постановка задачи**

Определить пользовательский класс. Определить в класс следующие конструкторы: без параметров, с параметрами, копирования. Определить в классе деструктор.



**Анализ задачи**

**1.** Для решения задачи необходимо:

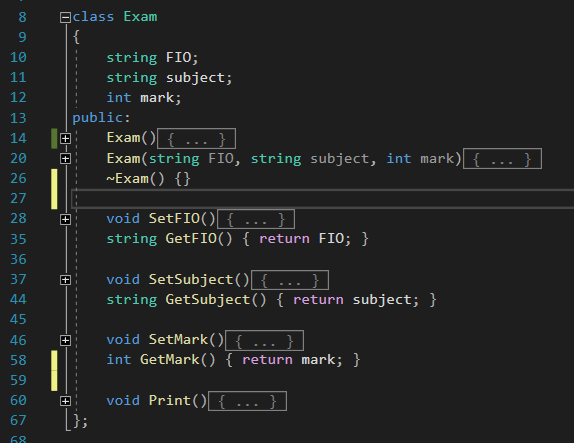
**1.1.** Организовать класс Exam для хранения ФИО в типе string, Предмета типа string и оценки в формате int.

**1.2.** Организовать необходимые методы для ввода данных в поля FIO и Subject, Mark: гетторы, сетторы, конструкторы, деструктор.

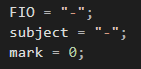
**1.3.** В функции main() использовать множественный if() для выбора пользователем, как он(-а) хочет ввести данные (через конструктор без параметров, с параметрами и конструктор копирования).

**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Организовать класс Exam для хранения ФИО в типе string, Предмета типа string и оценки в формате int.



**2.2.** Переменные типа string: FIO, Subject. Переменные типа int: mark, где FIO – введённые инициалы человека, Subject – введенный предмет, mark – оценка человека.

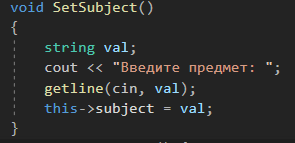
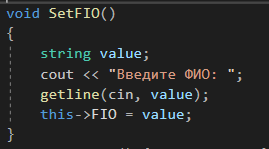


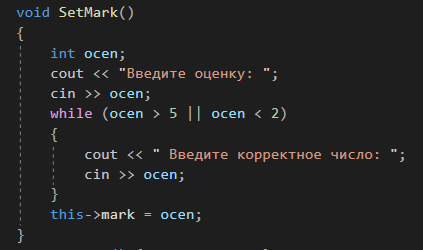
**2.3 .** STL-библиотекa vector используется для элементов типа Exam, чтобы выводить все объекты созданные различными способами.



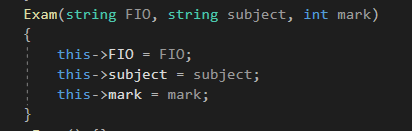
**3.** Для решения задачи данные были представлены в следующем виде:

**3.1.** Данные вводятся через объекты класса Exam через сетторы.



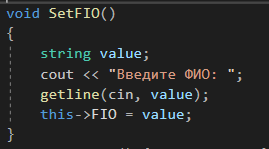


**3.2.** Данные также вводятся через объекты класса Exam через конструктор с параметрами.

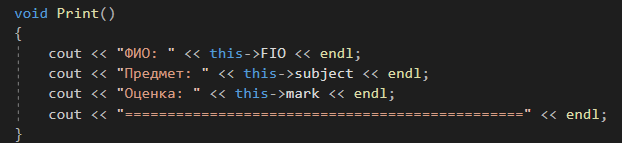


**4.** Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.** Для ввода числа используется сеттор (метод, описанный в классе для ввода), в котором используется функция cin.

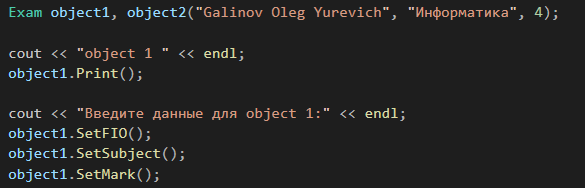


**4.2.** Для вывода данных на консоль используется метод Print().

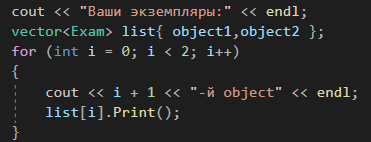


**5.** Поставленные задачи будут решены следующими действиями:

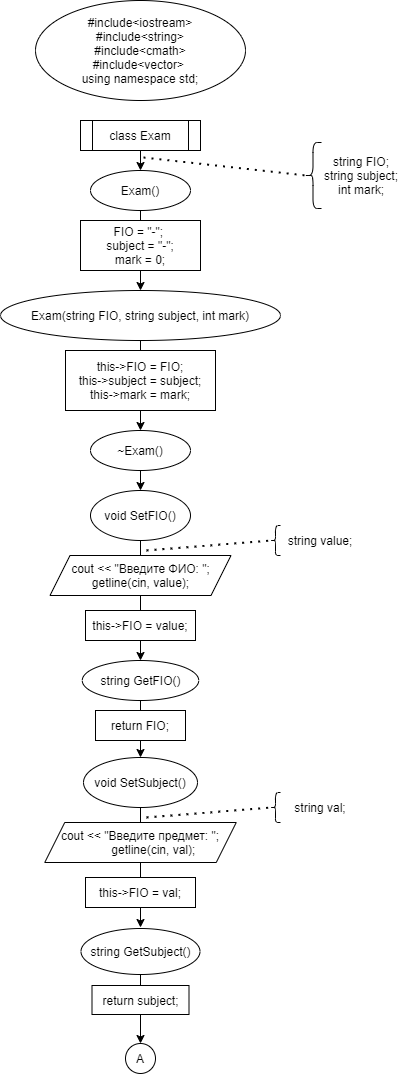
**5.1.** В функции main() создается два объекта класса Exam (с параметрами и без). Далее объекту без параметров присваиваются значения через описанные сеттеры, далее классы выводятся на экран через vector, с типом данных Exam.

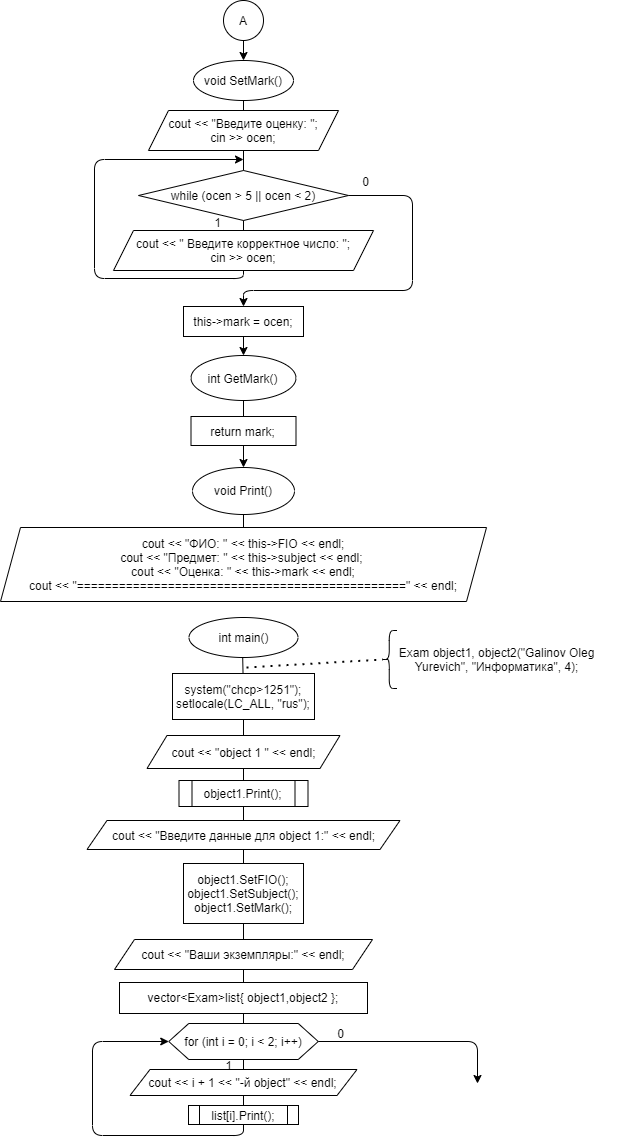


**5.2.** При помощи вектора происходит вывод всех элементов типа Exam.



**Блок-схема**

****



**Код**

#include <iostream>

#include <string>

#include <cmath>

#include <vector>

using namespace std;

class Exam

{

string FIO;

string subject;

int mark;

public:

Exam()

{

FIO = "-";

subject = "-";

mark = 0;

}

Exam(string FIO, string subject, int mark)

{

this->FIO = FIO;

this->subject = subject;

this->mark = mark;

}

~Exam() {}

void SetFIO()

{

string value;

cout << "Введите ФИО: ";

getline(cin, value);

this->FIO = value;

}

string GetFIO() { return FIO; }

void SetSubject()

{

string val;

cout << "Введите предмет: ";

getline(cin, val);

this->subject = val;

}

string GetSubject() { return subject; }

void SetMark()

{

int ocen;

cout << "Введите оценку: ";

cin >> ocen;

while (ocen > 5 || ocen < 2)

{

cout << " Введите корректное число: ";

cin >> ocen;

}

this->mark = ocen;

}

int GetMark() { return mark; }

void Print()

{

cout << "ФИО: " << this->FIO << endl;

cout << "Предмет: " << this->subject << endl;

cout << "Оценка: " << this->mark << endl;

cout << "===============================================" << endl;

}

};

int main()

{

system("chcp>1251");

setlocale(LC\_ALL, "rus");

Exam object1, object2("Galinov Oleg Yurevich", "Информатика", 4);

cout << "object 1 " << endl;

object1.Print();

cout << "Введите данные для object 1:" << endl;

object1.SetFIO();

object1.SetSubject();

object1.SetMark();

cout << "Ваши экземпляры:" << endl;

vector<Exam> list{ object1,object2 };

for (int i = 0; i < 2; i++)

{

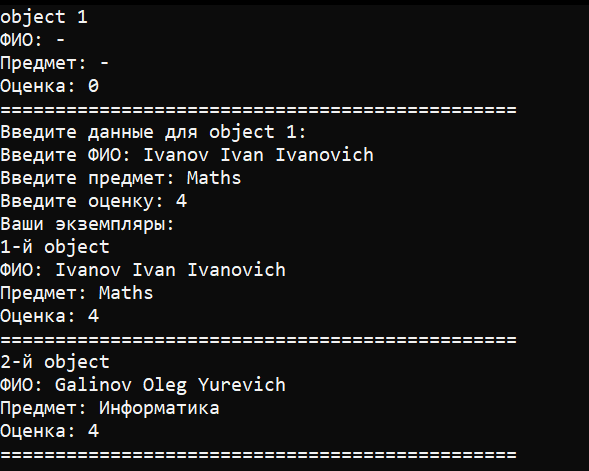
cout << i + 1 << "-й object" << endl;

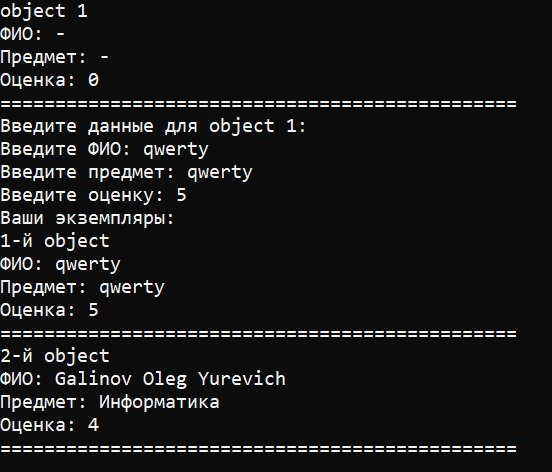
list[i].Print();

}

}

**Скриншоты**



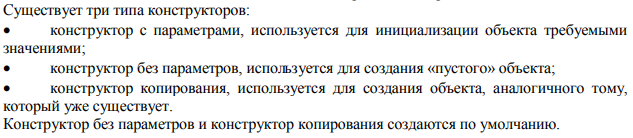


**Контрольные вопросы**



1. Конструктор необходим для инициализации объекта.



2. 

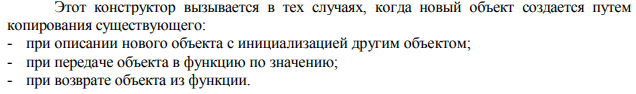


3. Деструктор – это особый вид метода, применяющийся для освобождения ресурсов, выделенных конструктором объекту. Описывать деструктор явным образом требуется в случае, когда объект содержит указатели на память, выделяемую динамически.



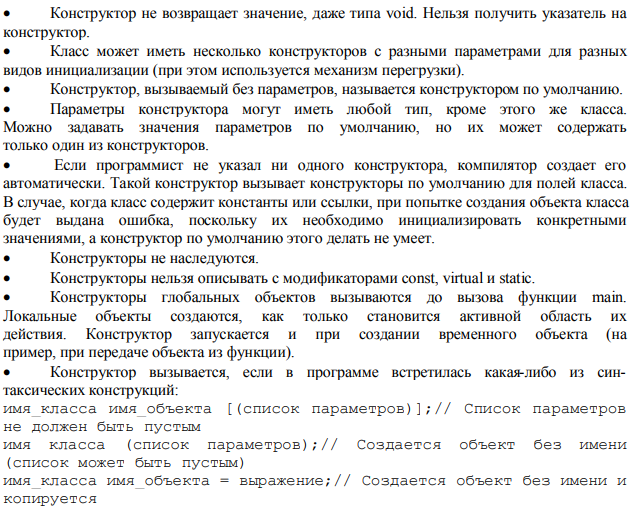
4. Конструктор с параметрами используется для инициализации объекта требуемыми значениями. Конструктор без параметров используется для создания «пустого» объекта. Конструктор копирования – это специальный вид конструктора, получающий в качестве единственного параметра указатель на объект этого же класса.



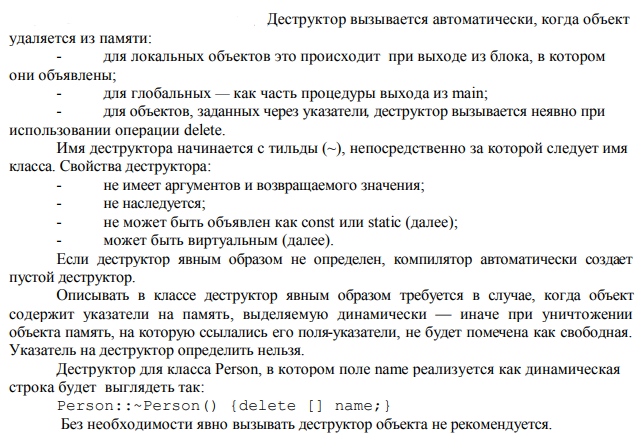
5. 



6.



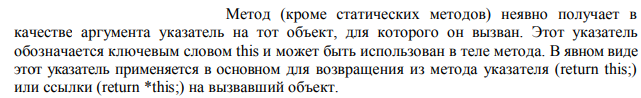


7. 

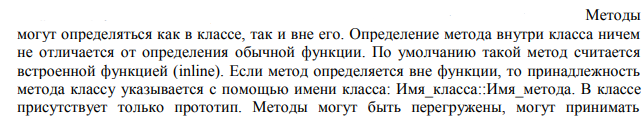


8.



9.



10.



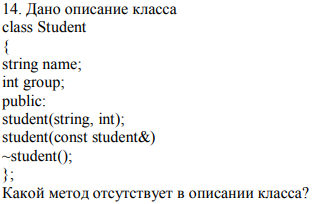
11. Конструктор не возвращает значения.



12. Конструкторы и деструкторы без параметров, если таковые не были прописаны явно.



13. Деструктор не возвращает значения.



14. Отсутствует конструктор без параметров в описании класса.



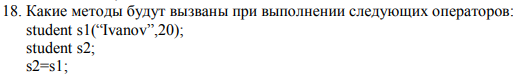
15. Будет вызван конструктор без параметров.



16. Будет вызван конструктор с параметрами.



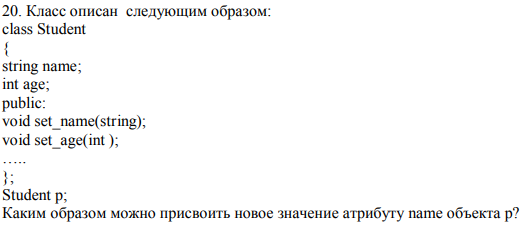
17. Будут вызваны конструктор с параметрами, а затем конструктор копирования.



18. Будут вызваны конструктор с параметрами, затем конструктор без параметров, а затем конструктор копирования.



19. Конструктор без параметров.



20. C помощью метода set\_name().