

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Пермский национальный исследовательский политехнический университет
Электротехнический факультет
Кафедра информационных технологий и автоматизированных систем

ОТЧЕТ
о работе по информатике
Семестр: 2

На тему: «Классы и объекты. Использование конструкторов.»

Выполнил студент ИВТ-22-26:
Дияров Тимур Артурович

Проверил доцент кафедры ИТАС:
Полякова Ольга Андреевна

Пермь 2023

Постановка задачи

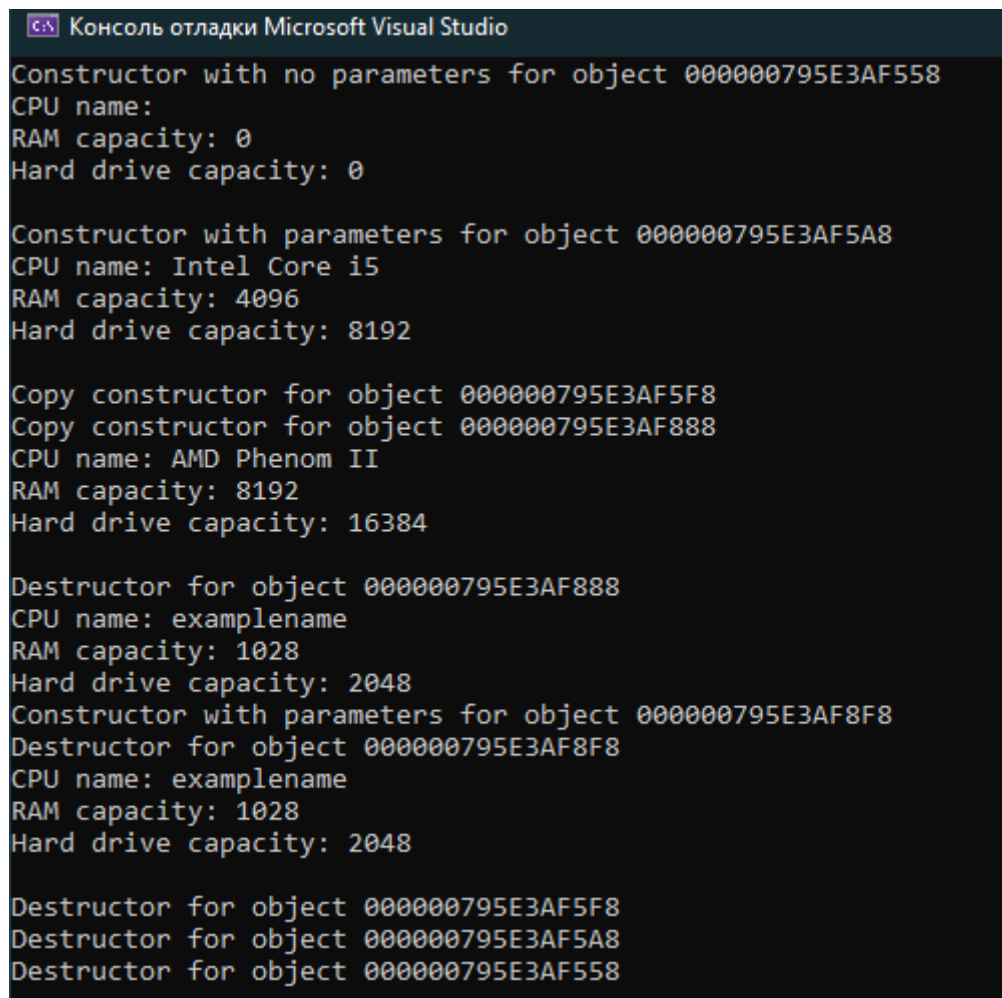
Реализовать пользовательский класс КОМПЬЮТЕР с переменными для процессора и объёма ОП с ЖД.

Описание структуры

Структура “fraction” с двумя переменными “first” (типа double) и “second” (типа int). В структуре имеются три функции типа void:

- Init(double, int): инициализация переменных с помощью двух чисел
- Read(): ручной ввод двух переменных
- Show(): вывод двух переменных в консоль

Скриншот работы программы



```
Консоль отладки Microsoft Visual Studio
Constructor with no parameters for object 000000795E3AF558
CPU name:
RAM capacity: 0
Hard drive capacity: 0

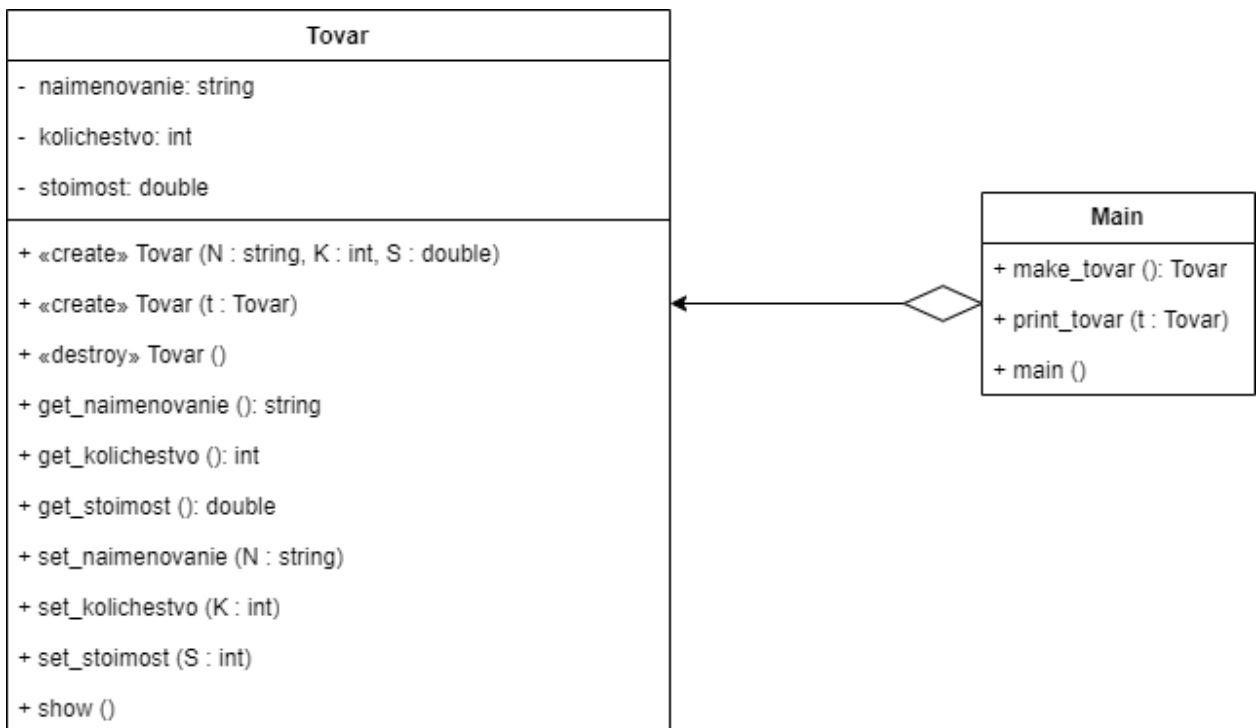
Constructor with parameters for object 000000795E3AF5A8
CPU name: Intel Core i5
RAM capacity: 4096
Hard drive capacity: 8192

Copy constructor for object 000000795E3AF5F8
Copy constructor for object 000000795E3AF888
CPU name: AMD Phenom II
RAM capacity: 8192
Hard drive capacity: 16384

Destructor for object 000000795E3AF888
CPU name: examplename
RAM capacity: 1028
Hard drive capacity: 2048
Constructor with parameters for object 000000795E3AF8F8
Destructor for object 000000795E3AF8F8
CPU name: examplename
RAM capacity: 1028
Hard drive capacity: 2048

Destructor for object 000000795E3AF5F8
Destructor for object 000000795E3AF5A8
Destructor for object 000000795E3AF558
```

UML-диаграмма



Контрольные вопросы

1. Для чего нужен конструктор?
Конструктор используется для инициализации объектов класса.
2. Сколько типов конструкторов существует в C++?
Конструктор без параметров, конструктор с параметрами и конструктор копирования.
3. Для чего используется деструктор? В каких случаях деструктор описывается явно?
Деструктор используется для освобождения ресурсов, выделенных объекту во время его жизни.
4. Для чего используется конструктор без параметров? Конструктор с параметрами? Конструктор копирования?
Конструктор без параметров используется для создания объекта без передачи ему каких-либо начальных значений.
5. В каких случаях вызывается конструктор копирования?
Конструктор копирования вызывается при создании нового объекта, который является копией существующего объекта.

6. Перечислить свойства конструкторов.
 - Имя конструктора совпадает с именем класса.
 - Конструктор не имеет возвращаемого типа.
 - Конструктор может быть перегружен, то есть класс может иметь несколько конструкторов с разными параметрами.
 - Конструктор может вызывать другие конструкторы внутри себя, используя инициализацию членов.
7. Перечислить свойства деструкторов.
 - Имя деструктора совпадает с именем класса, предваряется символом тильда (~).
 - Деструктор не имеет параметров.
 - Деструктор вызывается автоматически при удалении объекта или выходе из его области видимости.
 - Деструктор не может быть перегружен и не может вызывать другие деструкторы.
8. К каким атрибутам имеют доступ методы класса?

Методы класса имеют доступ ко всем атрибутам класса, включая приватные и защищенные.
9. Что представляет собой указатель this?

Указатель this представляет собой указатель на текущий объект класса.
10. Какая разница между методами определенными внутри класса и вне класса?

Методы, определенные внутри класса, являются методами-членами класса и имеют доступ ко всем членам класса, а методы, определенные вне класса, являются методами-друзьями класса и имеют доступ только к публичным членам класса.
11. Какое значение возвращает конструктор?

Конструктор не возвращает никакого значения.
12. Какие методы создаются по умолчанию?

По умолчанию создаются конструктор без параметров, конструктор копирования, оператор присваивания и деструктор.
13. Какое значение возвращает деструктор?

Деструктор не возвращает никакого значения.

14. Дано описание класса

```
class Student
{
    string name;
    int group;
public:
    student(string, int);
    student(const student&)
    ~student();
};
```

Какой метод отсутствует в описании класса?

В описании класса отсутствует метод определения имени студента.
С ним код может выглядеть так:

```
{
    string name;
    int group;
public:
    set_name(string)
    student(string, int);
    student(const student&)
    ~student();
};
```

15. Какой метод будет вызван при выполнении следующих операторов:

```
student*s;
s=new student;
```

Будет вызван конструктор класса Student без параметров.

16. Какой метод будет вызван при выполнении следующих операторов:

```
student s("Ivanov",20);
```

Будет вызван конструктор класса Student с параметрами.

17. Какие методы будут вызваны при выполнении следующих операторов:

```
student s1("Ivanov",20);
student s2=s1;
```

Будет вызван конструктор копирования класса Student.

18. Какие методы будут вызваны при выполнении следующих операторов:
- ```
student s1("Ivanov",20);
student s2;
s2=s1;
```

Будет вызван оператор присваивания класса Student.

19. Какой конструктор будет использоваться при передаче параметра в функцию print():
- ```
void print(student a)  
{a.show();}
```

Будет вызван конструктор копирования класса Student.

20. Класс описан следующим образом:

```
class Student  
{  
    string name;  
    int age;  
public:  
    void set_name(string);  
    void set_age(int ); ..... };  
Student p;
```

Каким образом можно присвоить новое значение атрибуту name объекта p?

Чтобы присвоить новое значение атрибуту name объекта p, можно использовать метод set_name(string):

```
main()  
{  
    p.set_name("новое значение");  
}
```