Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа по ООП**

**«№2»**

Выполнил:

студент группы РИС-23-2б

Ившин Максим Сергеевич

Проверила:

доцент кафедры ИТАС

О.А. Полякова

2024 г.

**Разработка алгоритма**

**Постановка задачи:**

1. Определить пользовательский класс.

2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.

3. Определить в классе деструктор.

4. Определить в классе компоненты-функции для просмотра и установки полей

данных (селекторы и модификаторы).

5. Написать демонстрационную программу, в которой продемонстрировать все три случая вызова конструктора-копирования, вызов конструктора с параметрами и конструктора без параметров.

**Анализ задачи:**

1. Создадим пользовательский класс Receipt.
2. Добавим конструктор без параметров, с параметрами (номер, дата, сумма), конструктор копирования. Также определим пустой деструктор.
3. Добавим геттер getInfo() и сеттер setInfo().

**Код программы на C++:**

main.cpp:

#include <iostream>

#include <string>

using namespace std;

#include "Receipt.h"

int main()

{

    // system("chcp 1251>null"); // для VS

    Receipt first\_rec;

    Receipt second\_rec(38234, "03.04.2024", 3000.5);

    Receipt third\_rec(second\_rec);

    first\_rec.getInfo();

    first\_rec.setInfo(1010102, "04.04.2024", 120.1823);

    first\_rec.getInfo();

}

Receipt.h:

#pragma once

#include <string>

#include <iostream>

using namespace std;

class Receipt

{

private:

    int number;

    string date;

    double sum;

public:

    Receipt(int, string, double);

    Receipt();

    ~Receipt();

    Receipt(Receipt &);

    void getInfo();

    void setInfo(int num, string date, double sum);

};

Receipt::Receipt()

{

    cout << "Вызвался стандартный конструктор\n";

    number = 0;

    date = "Unknown data";

    sum = 0;

}

Receipt::Receipt(int num, string date, double sum)

{

    cout << "Вызвался конструктор с параметрами:[number, date, sum]" << " = "

         << "[" << num << ", " << date << ", " << sum << "]\n";

    this->number = num;

    this->date = date;

    this->sum = sum;

}

Receipt::Receipt(Receipt &copying)

{

    cout << "Вызвался конструктор копирования\n";

    this->number = copying.number;

    this->date = copying.date;

    this->sum = copying.sum;

}

Receipt::~Receipt()

{

    cout << "Вызвался деструктор\n";

}

void Receipt::getInfo()

{

    cout << "Вызвался геттер: [number, date, sum]" << " = "

         << "[" << number << ", " << date << ", " << sum << "]\n";

}

void Receipt::setInfo(int num, string date, double sum)

{

    cout << "Вызвался сеттер с параметрами:[number, date, sum]" << " = "

         << "[" << num << ", " << date << ", " << sum << "]\n";

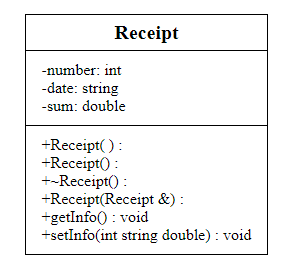
    this->number = num;

    this->date = date;

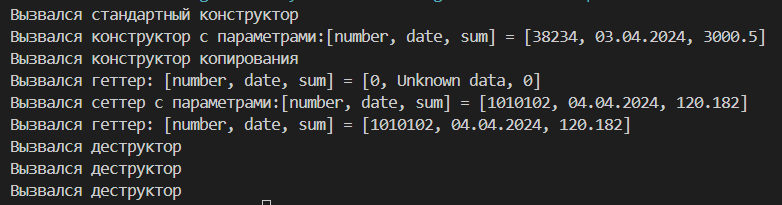
    this->sum = sum;

}

Диаграмма класса:



Пример работы программы:



**Контрольные вопросы**

1. **Для чего нужен конструктор?**

Конструкторы используются для создания экземпляров класса и инициализации их данных.

1. **Сколько типов конструкторов существует в С++?**

* С++ существует несколько типов конструкторов:
  + Конструктор без параметров (или конструктор по умолчанию)
  + Конструктор с параметрами, который позволяет передавать значения при создании объекта
  + Конструктор копирования, который используется при копировании

объектов

1. **Для чего используется деструктор? В каких случаях деструктор описывается явно?**

Деструктор используется для освобождения ресурсов, занятых объектом, таких как память, файлы или соединения с базой данных. Если конструктор описан явно, то необходимо и деструктор описать явно.



1. **Для чего используется конструктор без параметров?**

**Конструктор с параметрами? Конструктор копирования?**

Конструктор без параметров используется для инициализации объекта с начальными значениями по умолчанию. Конструктор с параметрами позволяет задать конкретные значения при создании объекта. Конструктор копирования используется для создания копии существующего объекта.

1. **В каких случаях вызывается конструктор копирования?**

Конструктор копирования вызывается, когда необходимо создать копию объекта, например, при присваивании одного объекта другому.

1. **Перечислить свойства конструкторов.**

Свойства конструкторов:

* + Может иметь параметры для инициализации объекта
  + Может быть перегружен для поддержки различных форматов инициализации
  + Вызывается автоматически при создании объекта

1. **Перечислить свойства деструкторов.**

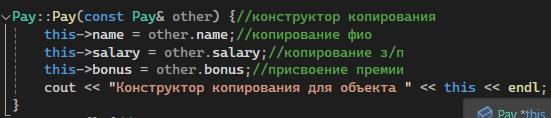
* Вызывается автоматически при уничтожении объекта
* Не может быть перегружен
* Не принимает параметров
* Не возвращает значения
* Не может быть объявлен как const , volatile или static

1. **К каким атрибутам имеют доступ методы класса?**

Private, Public, Protected

1. **Что представляет собой указатель this?**

Указатель this указывает на текущий объект, к которому обращается метод класса. Он используется внутри методов для доступа к данным объекта и вызова других методов того же класса.



1. **Какая разница между методами, определенными внутри класса и вне класса?**

Разница между методами, определенными внутри класса и вне класса, заключается в области видимости и доступе. Методы, определенные внутри класса, являются частью определения класса и могут быть вызваны только через объекты этого класса; можно не передать параметры. Методы, определенные вне класса, могут быть статическими и не требуют создания экземпляра класса для их вызова; передача параметров обязательна.

1. **Какое значение возвращает конструктор?**

Конструктор не возвращает значение, так как его цель - создание объекта, а не возврат результата.

1. **Какие методы создаются по умолчанию?**

Конструктор и деструктор.

1. **Какое значение возвращает деструктор?**

Деструктор не возвращает значение, так как его цель - освободить ресурсы, связанные с объектом, а не возвращать результат.

1. **Дано описание класса class Student**

**{**

**string name; int group; public:**

**student(string, int); student(const student&)**

**~student();**

**};**

**Какой метод отсутствует в описании класса?**

Конструктор по умолчанию.

1. **Какой метод будет вызван при выполнении следующих операторов: student\*s;**

**s=new student;**

Конструктор по умолчанию.

1. **Какой метод будет вызван при выполнении следующих операторов: student s(“Ivanov”,20);**

Конструктор с параметрами.

1. **Какие методы будут вызваны при выполнении следующих операторов: student s1(“Ivanov”,20);**

**student s2=s1;**

Конструктор с параметрами для первого объекта и конструктор копирования для второго.

1. **Какие методы будут вызваны при выполнении следующих операторов: student s1(“Ivanov”,20);**

**student s2; s2=s1;**

Конструктор с параметрами для первого объекта, конструктор без

параметров для второго объекта, конструктор копирования для второго

объекта.

1. **Какой конструктор будет использоваться при передаче параметра в функцию print(): void print(student a)**

**{a.show();}**

Конструктор по умолчанию.

1. **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 р?**

p.set\_name(“name”);

**GitHub:** *https://github.com/geroineee/PNRPU*