Министерство образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

"Пермский национальный исследовательский политехнический университет"

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Дисциплина: Основы алгоритмизации и программирования

Тема: Классы и объекты. Использование конструкторов.

Вариант: 15

|  |  |
| --- | --- |
| Выполнил работу | |
| Студент группы РИС-22-1б | |
| Мерзляков В.Н. | |
|  | |
| Проверил работу | |
| Доцент кафедры ИТАС | |
| Полякова О.А. | |
|  | |

Пермь – 2023

**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.

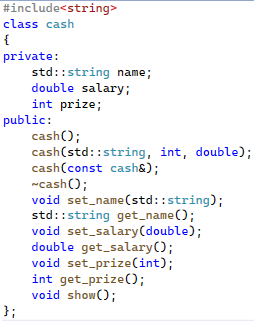
Вариант 15:

* Пользовательский класс ЗАРПЛАТА ФИО – string
* Оклад – double
* Премия (% от оклада) – int

Реализовать:

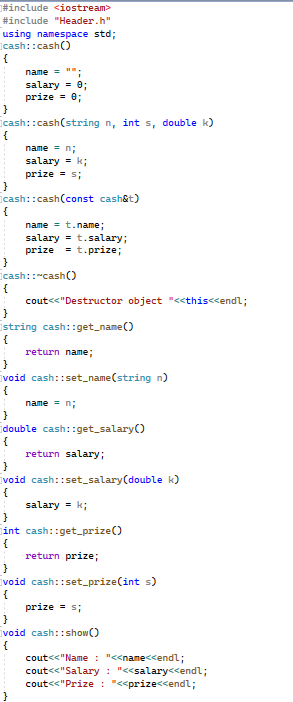
1. Вычитание пар чисел
2. Добавление константы к паре (увеличивается первое число, если константа целая, второе, если константа вещественная).

**Описание классов**

****

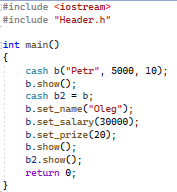
*Приложение 1 – Описание классов*

**Определение компонентных функций**

****

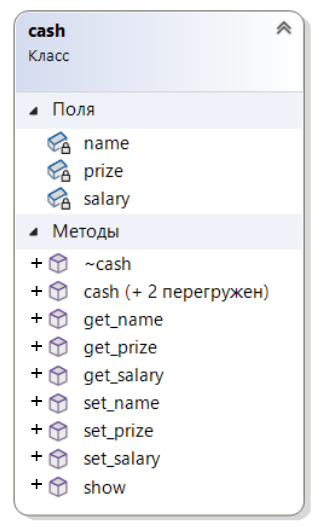
*Приложение 2 – Определение компонентных функций*

**Функция main()**

****

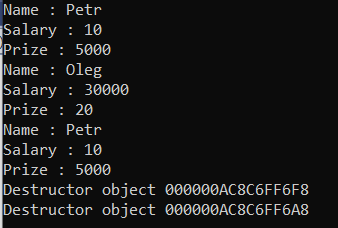
*Приложение 3 – Главная функция*

**UML-диаграмма**



*Приложение 4 – UML-диаграмма*

**Работа программы**



*Приложение 5 – Работа программы*

**Ответы на контрольные вопросы**

1. Конструктор нужен для инициализации объекта класса.

2. В С++ существует три типа конструкторов: конструктор по умолчанию, конструктор с параметрами и конструктор копирования.

3. Деструктор используется для освобождения памяти, выделенной для объекта класса. Он описывается явно, если класс имеет динамически выделенные ресурсы (например, указатели), которые необходимо освободить при удалении объекта.

4. Конструктор без параметров используется для создания объекта с некоторым начальным состоянием по умолчанию. Конструктор с параметрами используется для создания объекта с заданными параметрами. Конструктор копирования используется для создания копии объекта.

5. Конструктор копирования вызывается при создании копии объекта (например, при передаче объекта по значению).

6. Свойства конструкторов: имеют то же имя, что и класс; не имеют возвращаемого значения; могут быть перегружены; могут быть вызваны явно или неявно; вызываются автоматически при создании объекта.

7. Свойства деструкторов: имеют то же имя, что и класс, с префиксом ~; не имеют параметров; не имеют возвращаемого значения; вызываются автоматически при удалении объекта.

8. Методы класса имеют доступ к атрибутам объекта класса.

9. Указатель this представляет собой адрес объекта, для которого вызывается метод класса.

10. Методы, определенные внутри класса, являются членами класса и могут иметь доступ к его атрибутам. Методы, определенные вне класса, не являются членами класса и не имеют доступа к его атрибутам без использования параметров или указателей на объект класса.

11. Конструктор не возвращает значения, он создает объект класса.

12. По умолчанию создаются конструктор по умолчанию, конструктор копирования, оператор присваивания и деструктор.

13. Деструктор не возвращает значения, он освобождает выделенную память.

14. Отсутствует метод, который позволяет получить или изменить значение переменной name.

15. Будет вызван конструктор класса student.

16. Будет вызван конструктор класса student с параметрами "Ivanov" и 20.

17. При выполнении оператора student s1("Ivanov", 20); будет вызван конструктор класса student с параметрами "Ivanov" и 20, создающий объект s1.

При выполнении оператора `student s2 = s1; будет вызван конструктор копирования класса student, который скопирует значения полей объекта s1 в объект s2. Если конструктор копирования не был определен явно, то будет использован конструктор копирования по умолчанию, который копирует все поля объекта s1 в объект s2.

18. При выполнении оператора `student s1("Ivanov", 20); будет вызван конструктор класса student с параметрами "Ivanov" и 20, создающий объект s1.

При выполнении оператора student s2; будет вызван конструктор по умолчанию класса student, создающий объект s2 без параметров.

При выполнении оператора `s2 = s1;` будет вызван оператор присваивания класса student, который скопирует значения полей объекта s1 в объект s2. Если оператор присваивания не был определен явно, то будет использован оператор присваивания по умолчанию, который копирует все поля объекта s1 в объект s2.

19. Будет использоваться конструктор копирования класса student, так как в функцию передается объект класса student по значению.

20. Можно использовать метод set\_name(), передав ему новое значение:

p.set\_name("Новое имя");