

Университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.04.01 Информатика и вычислительная техника
Дисциплина «Программирование на C++»

Отчет По лабораторной работе №2

Выполнил:
Казаченко Р. О.

Преподаватель:
Лаздин А. В.

Санкт-Петербург, 2024 г.

Вариант задания:

Необходимо спроектировать класс, реализующий хранение данных, связанных с экземпляром класса в динамической памяти. Это может быть, например представление целых чисел, для которых помимо значения типа `int` хранится строковое представление числа (22 и «двадцать два»). Пример условен.

Для данного класса необходимо реализовать все необходимые конструкторы, включая конструкторы копирования и перемещения, деструктор. Все конструкторы и деструктор должны «сообщать» о своём вызове. `std::cout << "Copy constructor" << std::endl;` то же относится и к перегрузке операции присваивания (два варианта: без перемещения и с перемещением). Определение класса должно быть помещено в заголовочный файл (.hpp), реализация методов в отдельном файле .cpp.

Написать программу (ещё один модуль cpp) осуществляющую работу с экземплярами разработанного класса, которая должна демонстрировать:

- Создание статических и динамических экземпляра класса, стандартного вектора в стиле Си, написать функцию для обработки данного вектора;
- Передача экземпляров класса в функцию, и возврат экземпляра из функции; включая вариант с передачей и возвратом значений типа ссылки на класс;
- Продемонстрировать работу с разработанным классом, создавая вектора и списки экземпляров класса, используя стандартные классы `vector` и `list` длиной от 5 до 10.

Проанализировать процессы создания и удаления экземпляров класса для различных примеров. Необходимо получить вывод от всех конструкторов, деструктора и перегруженных операций присваивания.

Исходный код:

<https://github.com/R4MZEZ/SiPlusPlus>

Примеры работы:

Тест создания статических и динамических экземпляра класса:

```
--- STATIC TEST START ---  
Basic constructor 1  
Basic constructor 2  
Destructor 2  
Destructor 1  
--- STATIC TEST END -----
```

Тест создания вектора экземпляров класса:

```
--- VECTOR TEST START ---  
Basic constructor 3  
Move constructor 3 to 4  
Destructor 3  
Basic constructor 5  
Move constructor 5 to 6  
Basic constructor 7  
Copy constructor 4 to 7  
Destructor 4  
Destructor 5  
Basic constructor 8  
Move constructor 8 to 9  
Basic constructor 10  
Copy constructor 7 to 10  
Basic constructor 11  
Copy constructor 6 to 11  
Destructor 7  
Destructor 6  
Destructor 8  
Basic constructor 12  
Move constructor 12 to 13  
Destructor 12  
Basic constructor 14
```

```
Move constructor 14 to 15  
Basic constructor 16  
Copy constructor 10 to 16  
Basic constructor 17  
Copy constructor 11 to 17  
Basic constructor 18  
Copy constructor 9 to 18  
Basic constructor 19  
Copy constructor 13 to 19  
Destructor 10  
Destructor 11  
Destructor 9  
Destructor 13  
Destructor 14  
First vector element  
Second vector element  
Third vector element  
Fourth vector element  
Fifth vector element  
Destructor 16  
Destructor 17  
Destructor 18  
Destructor 19  
Destructor 15  
--- VECTOR TEST END -----
```

Тест создания списка экземпляров класса:

```
--- LIST TEST START -----  
Basic constructor 20  
Move constructor 20 to 21  
Destructor 20  
Basic constructor 22  
Move constructor 22 to 23  
Destructor 22  
Basic constructor 24  
Move constructor 24 to 25  
Destructor 24  
Basic constructor 26  
Move constructor 26 to 27  
Destructor 26  
Basic constructor 28
```

```
Move constructor 28 to 29  
Destructor 28  
First list element  
Second list element  
Third list element  
Fourth list element  
Fifth list element  
Destructor 21  
Destructor 23  
Destructor 25  
Destructor 27  
Destructor 29  
--- LIST TEST END -----
```

Тест передачи и возврата экземпляра класса в функцию по значению:

```
--- FUNC TEST START -----  
Basic constructor 30  
Old size: 4  
Basic constructor 31  
Copy constructor 30 to 31  
Move constructor 31 to 32  
Destructor 31  
New size: 4  
Destructor 32  
Destructor 30  
--- FUNC TEST END -----
```

Тест передачи и возврата экземпляра класса в функцию по ссылке:

```
--- REF FUNC TEST START -----  
Basic constructor 33  
Old size: 4  
New size: 999  
Destructor 33  
--- REF FUNC TEST END -----
```

Выводы:

Изучил механизм вызова конструкторов и деструкторов, а также механизм их вызова в различных сценариях обработки объектов.