

**Дипломная работа студента 541 группы
Марии Крень**



Пользовательский уровень библиотеки неконсервативной сборки мусора для C++

Научный руководитель: к.ф.-м.н. Д.Ю. Булычев

Рецензент: к.ф.-м.н. Д.В. Кознов

Введение

- **Сборка мусора (англ. Garbage collection)** – универсальное средство управления динамической памятью.
- Язык программирования C++ не совместим со сборкой мусора. Содержит низкоуровневые конструкции оперирования с памятью.



Цели дипломной работы

- создание пользовательского уровня для библиотеки неконсервативной сборки мусора для C++

Задачи

- изучить и проанализировать существующие подходы к автоматизации и управлению памятью C++
- предложить и реализовать собственное решение на основе проведенного анализа
- продемонстрировать работоспособность реализованного сборщика мусора

Некоторые средства автоматизации управления памятью для C++

- **“Умный указатель”** – специальный объект, который хранит указатель на участок динамически отведенной памяти и определяет некоторую дисциплину обращения с этим указателем.
- **Недостаток:**
 - определенным образом ограничивается набор возможных способов манипулирования данными

Пользовательский уровень библиотеки сборки мусора

- Один шаблонный класс “умного указателя” – *gc_ptr*.

Часть операторов реализованных внутри данного класса:

- T& operator*() const
- T* operator->() const, operator T*() const
- T& operator[](size_t index)
- ...

- Одна шаблонная функция выделения памяти – *gc_new*.

Функция позволяет выразить все ситуации, в которых может быть употреблен оператор выделения памяти **new**.

Пример использования пользовательских примитивов(1)

```
class GCString {  
    private:  
        gc_ptr<char> pData;  
        int length;  
        GCString (gc_ptr<char> p, int l) : pData (p), length (l) {};  
    public:  
        GCString () : length (0), pData (NULL) {};  
        GCString (const char *cString);  
        virtual ~GCString () {};  
        GCString (const GCString &s);
```

Пример использования пользовательских примитивов(2)

```
GCString GCString::operator+ (const GCString &s) {  
    gc_ptr<char> p = gc_new<char> (length + s.length + 1);  
    strcpy (p, (char*) this->pData);  
    strcat (p, (char*) s.pData);  
    return GCString (p, length + s.length);  
}
```

```
GCString GCString::operator+ (const char *cString) {  
    return *this + GCString (cString);  
}
```

Результаты

- Изучены различные способы автоматизации управления памятью для языка C++ на основе использования **“умных указателей”**
- Разработан интерфейс библиотеки, позволяющей реализовать неконсервативную сборку мусора при выполнении определенных соглашений
- Полученная библиотека была протестирована на ряде примеров