

Семинар 1. Класс «Точка»

С. А. Шершаков

10 сентября 2018 г.

В документе представлен 1-й цикл заданий для самостоятельного выполнения студентами курса «Алгоритмы и структуры данных» и методические рекомендации по их выполнению. Задание посвящено изучению основных конструкций языка C++, а также особенностям построения проектов для разных платформ и разными средствами сборки на основе единой кодовой базы. Также рассматриваются вопросы unit-тестирования C++-программ с помощью библиотеки gtest.

Ред. 1.2 от 10.09.2018 г.

1 Требования, цели и ожидаемые результаты

1.1 Требования

Студенты должны владеть следующими знаниями, умениями и навыками для выполнения задания.

- Основы разработки программ на языке C++.
- Основы системы CMake построения C++-программ или IDE Microsoft Visual Studio.
- Методические рекомендации к практическим занятиям по дисциплине «Алгоритмы и структуры данных».

Чтобы перейти непосредственно к шагам, необходимым для выполнения задания, обратитесь к разделу 4.

1.2 Цели и задачи

Основная цель работы — овладение основными языковыми и инструментальными подходами к написанию программ на языке C++; рассмотрение основных отличий разработки на языке C++ от языков C# и Java.

Основные темы: принципы создания C++-проектов с помощью инструментов построения (toolchain) gcc и CMake; модули языка, заголовочные файлы и модули трансляции; базовая система ввода/вывода; основные конструкции языка; память и указатели.

Связанные задачи: unit-тестирование с использованием библиотеки gtest¹.

Задачи для выполнения в рамках самостоятельной работы:

- Изучение заголовочных файлов классов Point и PointArray (модуль xipoint.h).
- Реализация объявленных методов классов Point и PointArray в модуле xipoint.cpp на основе спецификации методов, ориентируясь на их спецификацию и настоящее руководство.
- [Опционально] протестировать полученную реализацию на множестве предложенных unit-тестов².

¹ <https://github.com/google/googletest>

² Студенты получают часть unit-тестов (30–40 % от полного набора).

1.3 Ожидаемые результаты

Студенты, успешно³ выполнившие задание, получают навыки разработки C++-программ, включающих базовые управляющие структуры, массивы, разделение модулей на интерфейсную и реализационную части, использование препроцессора языка, тестирования разработанных методов на предложенном наборе unit-тестов.

³ И самостоятельно.

2 Описание задания

В рамках самостоятельной работы студентам предлагается разработать недостающие методы классов `Point` и `PointArray`, интерфейсное описание и спецификация которых представлены в файле `xipoint.h`. Разработанные методы помещаются в подготовленный файл `xipoint.cpp`.

Далее представлена краткая информация по основным модулям/типам программного кода. Дополнительная информация может быть получена из комментариев к файлам/типам.

2.1 Класс `Point`

Класс `Point` описывает точку с двумя целочисленными координатами, хранимыми в защищенных полях. Класс содержит конструктор инициализации обеих координат, получающий параметры по умолчанию, что позволяет его использовать только для одной координаты x либо вообще инициализировать без параметров, то есть использовать в качестве конструктора по умолчанию.

Класс содержит методы для установки и получения значений отдельных координат (с перегруженными константными вариантами).

2.2 Класс `PointArray`

Представляет собой массив точек, представленных объектами класса `Point`. Класс самостоятельно управляет памятью подлежащего C-массива, предоставляя пользователю интерфейс для добавления, удаления индивидуальных точек, получения доступа к ним в виде указателей. Размер массива — число хранимых объектов — доступен через вызов метода `getSize()`.

Класс содержит несколько конструкторов, позволяющих по-разному конструировать объект. При каждом изменении размера подлежащего массива (добавление и удаление элементов) вызывается служебный (защищенный) метод `resize()`⁴. Очищение массива осуществляется методом `clear()`. При удалении объекта деструктор освобождает память, занимаемую массивом, осуществляя тем самым политику избегания «утечки памяти».

⁴ Данный метод создает новый массив запрашиваемой длины, копирует в него элементы из предыдущего массива (максимально возможное их число), удаляет предыдущий массив и перезаписывает указатели на массив и его размер. Метод осуществляет возможно неоптимальное перераспределение памяти при каждом изменении массива, однако в рамках данной задачи оптимальность не является изучаемым критерием.

3 Структура проекта

Студенты получают для работы проект со следующей структурой:

- `/docs` — каталог с документацией, включающий настоящее руководство;
- `/src` — исходные коды проекта;
- `/tests/gtest` — тесты (исходные файлы) в формате gtest;
- `/tests/gtest/gtest` — исходные файлы системы unit-тестирования gtest;
- `/readme.md` — самая общая и самая полезная дополнительная информация по проекту.

Проект поставляется в виде одного или нескольких (возможно заархивированных) файлов, что определяется конкретным способом представлением задачи с использованием автоматизированной системы обучения (LMS).

4 Указания по выполнению задания

Выполнение задачи сводится к следующей последовательности⁵.

- Ознакомиться с настоящим руководством-заданием.
- Получить исходные файлы для работы⁶.
- Ознакомиться с предлагаемыми для работы исходными файлами.
- Подготовить реализацию недостающих методов⁷ классов `Point` и `PointArray` в файле `/src/xipoint.cpp`.
- Протестировать (по желанию) реализацию на предложенном наборе тестов (файлы `/tests/point_test.cpp` и `point_array_test.cpp`).
- Загрузить результаты работы в виде единого файла `xipoint.cpp` в ассоциированный проект системы LMS до крайнего срока, указанного в сводке проекта.

При выполнении задачи не подразумевается изменение каких-либо файлов, не перечисленных выше.

4.1 Сдаваемые файлы

- `/src/xipoint.cpp` — реализация основной задачи.

⁵ Рекомендованная последовательность.

⁶ С использованием системы LMS:
<http://lms.hse.ru>

⁷ Важно: проект с исходным набором файлов **не будет** собираться по причине отсутствия реализации некоторых задекларированных методов, — подлежащий реализации в рамках данной работы.