**Задание 1**

Реализовать на языке С++ одномерный массив (в процедурной парадигме). Выделить необходимые для реализации функции (заполнение, вывод в строку, нахождение минимального (максимального, среднего) или сортировка).

Для каждой функции написать документацию, придерживаясь правилам, описанным в литературе [1].

Задание выполнять в системе контроля версий [2]. Проверка задания осуществляется посредством pull request в веб-хостинге GitHub (collaborator MarinaPaley). Задание считается зачтенным, если получен approve.

**Задание 2**

Разработать структуру данных на языке программирования С++ в ООП парадигме. Реализовать библиотеку классов, обеспечить выполнение операций CRUD (create, read, update, delete) для любого типа данных (template). Переопределить операторы сдвига влево и вправо для возможности чтения и записи из любого потока (консоль, строка, файл). Ни один метод не должен зависеть от консоли или файла, используйте представление в строку. Написать итератор для работы с коллекцией (разработанной структурой данных). По возможности использовать умные указатели (smart pointer), заместо «сырых» (raw). Использование стандартных структур данных возможно в качестве вспомогательных, например, при обходе дерева можно использовать std::vector. Для избежания недоразумений, лучше обсудить использование с преподавателем индивидуально.

Все методы покрыть тестами (модульное тестирование).

Задание выполнять в системе контроля версий. Проверка задания осуществляется посредством pull request в веб-хостинге GitHub (collaborator MarinaPaley). В репозитории необходимо иметь рабочий код, так, чтобы для проверки преподаватель смог склонировать репозиторий и запустить тесты, не прибегая к помощи специальных инструментов (если разрабатываете в MS Visual Studio, то репозиторий должен содержать решение и минимум три проекта: запускаемый, библиотеку классов, тесты, если разрабатываете в другой IDE, то необходимо приложить cmake файлы для запускаемого и тестируемого проектов). Использовать для решения последнюю версию стандарта языка С++ (20).

Задание считается зачтенным, если получен approve.

Варианты 1–7 для максимальной оценки «удовлетворительно» на экзамене.

**Рекомендации**. Начинать работу над структурой данных лучше с API для конкретного типа данных, например int. После подтверждения преподавателем минимального набора функционала, необходимо реализовать все объявленные методы и отладить на них тесты. Разработку шаблонного класса выбранной коллекции удобно вести, применяя TDD (test driven development) [3], так как необходимые классы для целого типа данных вами уже отлажены. Необходимо будет «скопировать» логику тестов для любого ссылочного типа данных, например строки, и уже имея готовую тестовую базу, разработать библиотеку на шаблонах.

Все члены классов должны иметь документацию в стиле doxygen.

Необходимо предоставить отчет о работе, содержащий UML диаграмму библиотеки классов.

**Структуры данных**

1. Стек
2. Дек
3. Очередь
4. Односвязный список
5. Двусвязный список
6. Односвязный циклический список
7. Двусвязный циклический писок
8. Очередь с приоритетом
9. Бинарное дерево поиска
10. АВЛ-дерево
11. Красно-черное дерево
12. Дерево отрезков
13. Граф
14. Словарь (map)
15. Хэш-таблица (hash table)
16. T (trie) дерево
17. Хэш-сет (hash set)
18. Куча (heap)
19. Ассоциативный массив
20. Splay-дерево
21. Матричное дерево
22. Декартово дерево

# Список литературы

x

|  |  |
| --- | --- |
| 1. | Documenting C++ Code¶ [Электронный ресурс] // Association of Universities for Research in Astronomy, Inc. (AURA).: [сайт]. [2023]. URL: https:/​/​developer.lsst.io/​cpp/​api-docs.html#documentation-must-use-javadoc-style-tags (дата обращения: 21.09.2023). |
| 2. | Васильева М.А., Филипченко К.М. Система контроля версия. Основы командной разработки: учебное пособие для ВУЗов. Санкт-Петербург: Лань, 2022. 144 с. |
| 3. | Кент Б. Экстремальное программирование. Разработка через тестирование TDD. Спб.: Питер, 2020. 224 с. |

x