

Лабораторная работа №3

Объектно-ориентированное программирование

Цели и задачи работы: изучение основных принципов объектно-ориентированного программирования и основ юнит-тестирования.

Задание к работе: Самостоятельно решить задачи в соответствии с индивидуальным вариантом.

Методика выполнения работы:

1. Разработать алгоритмы решения задачи по индивидуальному заданию.
2. Написать и отладить программы решения задачи (C++, Go или Rust).
3. Протестировать работу программ на различных исходных данных.
4. По запросу преподавателя быть готовым модифицировать/добавить алгоритмы/блоки кода в контексте ООП.
5. Ответить на теоретические вопросы к лабораторной работе на выбор преподавателя (не менее двух вопросов).

Перечень вопросов к защите лабораторной работы 3.

1. Классы. Назначение секций класса. Конструкторы, списки инициализации.
2. Деструкторы и время жизни объектов класса. Константность методов.
3. Классы. Порядок конструирования объектов класса. Шаблоны классов
4. Хэш функции. Понятие. Свойства хэш функций. Требования к хэш функциям. Понятие коллизии.
5. Элементарные способы построения хэш-функций. Классификация хэш-функций. Современные криптографические хэш-функции.
6. Применение хэш-функций в криптографии.
7. Основные принципы создания объектной модели (абстрагирование, инкапсуляция, модульность, иерархия, типизация, параллелизм, сохраняемость).
8. Экземпляры класса. Уровни доступа к членам класса. Классификация методов объекта.
9. Средства UML для создания абстракций. Вид нотации класса, взаимодействия классов. Представление иерархических отношений.
10. Понятие интерфейса. Интерфейс и абстрактный класс: сравнение.
11. Наследование.
12. Полиморфизм. Понятие виртуальных методов. sizeof классов при наследовании (обоснование).
13. Сериализация и десериализация данных. Область применения. Форматы сериализации данных.
14. Понятие тестирования. Основные цели тестирования.
15. Стандарты тестирования, основные показатели.
16. Поддерживаемость программы на основе ISO 25010 и ее составные элементы.
17. Основные виды тестирования.
18. Основные принципы тестирования.
19. Уровни тестирования.

20. Как Вы считаете: почему не рекомендуют, чтобы тестированием занимались программисты?

21. Юнит тестирование в C++: Boost Test Library, Google test, CxxTest. Реализация

22. Юнит тестирования на примере другого ЯП (Go или Rust). Основные фреймворки.

Реализация абстрактных структур данных.

1. Реализовать классы¹ с базовым набором операций (private, public) на основе лабораторной работы 1 на одном из множества языков $X \in \{\text{Rust}, \text{Go}, \text{C}++\}$:
 - a. Массив
 - b. Список (односвязный, двусвязный).
 - c. Очередь
 - d. Стек
 - e. Хеш таблицы
 - f. Деревья (вариант 1 Бинарное дерево поиска, вариант 2 Full Binary Tree, вариант 3 Complete Binary Tree)
 2. Реализовать класс «Стек» с базовым набором операций (private, public) на основе лабораторной работы 1 на одном из множества языков $Y \in \{\text{Rust}, \text{Go}, \text{C}++\}/X$.
 3. Реализовать покрытие тестами (не менее 90%) в проектах 1 и 2. Benchmark использование обязательно.
 - a. C++: Boost Test Library, Google test, CxxTest
 - b. Go: testing, testify. Создать HTML-отчет о покрытии, который предоставляет визуальный анализ того, как много кода было протестировано.
 - c. Rust: Unit-Test Rust[Ⓜ]
 4. Выполнить сериализацию и десериализацию данных бинарного и текстового форматов для проектов 1 и 2 соответственно².
 5. В отчете представить UML диаграммы.
 6. Теоретическая часть (не более 15 страниц).
 - a. Четный вариант. Проанализировать мировые практики к оценке качества ПО (например, ISO 25010).
 - b. Нечетный вариант. Анализ российских стандартов оценки качества ПО.
 - c. Для всех вариантов. Основные виды тестирования.
- Отчет содержит список использованных источников.

¹ Обязательно использовать многофайловый проект.

² В случае трудоемкости процессов сериализации/десериализации на ЯП C++ допускается использование только бинарного формата

Литература

1. Майерс, С. Искусство тестирования программ / С. Майерс [и др.]. – М. : Диалектика, 2020. – 272 с.
2. Кент, Б. Экстремальное программирование: разработка через тестирование / Б. Кент. – СПб: Питер, 2023. – 224 с.
3. Назина, О. Что такое тестирование. Курс молодого бойца / О. Назина. – БХВ, 2022. – 592 с.

<https://habr.com/ru/articles/597859/>