

Лабораторные работы

Вместо лабораторных работ (кроме первой) можно разрабатывать собственный проект по договорённости с преподавателем. Индивидуальный проект должен предполагать самостоятельный процесс объектно-ориентированного проектирования.

Семестр I (2018-2019)

Задание 0. C++

???

Задание 1. Простой класс

Описать АДТ. Описать класс на C++. Реализовать методы для доступа и изменения данных, контроль постоянства, конструктор с параметрами. Операторы и генерирование исключительных ситуаций если необходимо.

Классы на выбор:

- Класс Время. Сложение, вычитание. Добавление минут, секунд, часов и т. п. Перевод времени в секунды, часы, минуты. Конвертирование в строку.
- Дата. Реализовать то же самое, что и для времени.
- Комплексное число.
- Кватернион?
- Вектор?
- Другой класс по согласованию с преподавателем.

Задание 2. Диаграмма классов

Составить UML диаграмму классов. Не менее 4 классов. Каждый взаимодействует хотя бы с одним другим. Взаимодействие не должно быть только последовательным. Объекты не должны быть однотипными. Должно быть как минимум по одному отношению: ассоциация, агрегация (композиция), наследование. Указать мощность связей. Диаграмму оформить в электронном и твёрдом формате. Сохранение диаграммы исключительно в формат растровых изображений не допускается. Презентация схемы на доске или проекторе. Обсуждение в группе.

Задание 3. Работа с Git.

1. Создать репозиторий.
2. Добавить файлы к отслеживанию.
3. Сделать коммит (зафиксировать изменения).
4. Исправить предыдущий коммит.
5. Создать новую ветку.
6. Переключиться на неё.
7. Внести изменения. Зафиксировать их.
8. Посмотреть что содержится в файлах на ветке master.
9. Объединить ветки.
10. Решить конфликт.
11. Клонировать удалённый репозиторий?
12. Отправить изменения в удалённый репозиторий?

Задание 4. Класс «матрица»

Создать класс представляющий матрицу. В качестве основы использовать класс vector.

Реализовать:

- доступ к отдельным элементам матрицы,
- доступ к строкам матрицы?
- сложение, вычитание
- умножение на число
- умножение матрицы на матрицу
- транспонирование,
- вычисление определителя,
- вычисление обратной матрицы.
- заполнение матрицы одним значением
- заполнение матрицы случайными числами,
- Создание диагональной матрицы.
- Операторы $\ast=$, $-$, $+=$?

Наглядно продемонстрировать работу всех методов. Недопустимые или невозможные операции над матрицами обрабатывать с помощью механизма генерации исключений.

Рекомендуется использование системы контроля версий при разработке.

Пример: QgenericMatrix - <http://doc.qt.io/qt-5/qgenericmatrix.html>

Задание 5. Реализовать диаграмму классов на C++

Реализовать предыдущую лабораторную работу. Как минимум одно поле и метод должны быть переопределены. Каждый класс должен располагаться в отдельном модуле.

Представить состояние и изменение объектов наглядно.

Рекомендации: вести разработку «сверху вниз», использовать систему контроля версий.

Необходимо знать: Описание классов в C++. Конструкторы. Методы. Простое наследование. Некоторые stl контейнеры (list, vector, string). Динамический полиморфизм.

Задание ?.

Матричный калькулятор с графическим интерфейсом пользователя?

Диаграмма классов на C++ с графическим интерфейсом пользователя?

Фильтрация ввода некорректных данных. Удобный UI.