

Лабораторные работы

Вместо лабораторных работ (кроме первой) можно разрабатывать собственный проект.

Семестр I

Задание. Диаграмма классов

Составить объектную схему 3-5+ объектов. Каждый взаимодействует хотя бы с одним другим. Взаимодействие не должно быть только последовательным. Объекты не должны быть однотипными. Должно быть как минимум одно отношение типа наследование как минимум одна агрегация. Схему оформить в электронном и твёрдом формате. Презентация схемы на доске или проекторе. Обсуждение в группе.

+ использование UML

Необходимо знать: ОО декомпозицию. Виды отношений между классами. UML диаграмму классов.

Задание. Реализовать диаграмму классов на C++

Реализовать предыдущую лабораторную работу. Возможны изменения.

~~Реализовать как минимум трёхуровневое наследование.~~ В качестве классов использовать модели из реального мира. В каждом наследнике должны быть добавлены новые поля и методы. Как минимум одно поле и метод должны быть переопределены.

Представить состояние объектов наглядно.

Рекомендации: Описывать для начала «заглушки» для методов.

Рекомендуется использование системы контроля версий при разработке.

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

Задание. 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.

Задание.

Матричный калькулятор с графическим интерфейсом пользователя?

Диаграмма классов на C++ с графическим интерфейсом пользователя?

Фильтрация ввода некорректных данных. Удобный UI.

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?

Семестр II

Задание.

Игра «Жизнь», Сапёр, Тетрис, свой вариант.

Задание

Эмуляции процессов какой-либо предметной области?

Задание

???

Задание

???