

Объектно-ориентированное программирование

Лабораторные работы (очная форма обучения)

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 1. Работа с Git.

Продemonстрировать работу с системой управления версиями git на примере одного из проектов (программ) созданных на лабораторных занятиях.

Сценарий использования для примера:

1. Создать репозиторий.
2. Добавить файлы к отслеживанию.
3. Сделать коммит (зафиксировать изменения).
4. Исправить предыдущий коммит.
5. Создать новую ветку.
6. Создать worktree для ветки master.
7. Переключится на неё.
8. Внести изменения. Посмотреть разницу (diff)
9. Зафиксировать изменения
10. Посмотреть что содержится в файлах на ветке master.
11. Объединить ветки.
12. Решить конфликт.
13. Клонировать удалённый репозиторий
14. Отправить изменения в удалённый репозиторий, забрать изменения из удалённого репозитория.

Вопросы.

Что такое система управления версиями? Для чего она используется?

Что такое репозиторий (локальный и удалённый)?

Какие файлы следует добавлять к отслеживанию, а какие нет?

В каких случаях создавать ветку?

Что такое конфликт? Как исправить?

Ссылки

- <http://uleming.github.io/gitbook/index.html>
- <https://inoyakaigor.ru/blog/85>

Задание 2. 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:

- 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.

Вопросы

Что такое АДТ?

Что такое класс? Что такое объект?

Что такое абстрагирование?

Что такое инкапсуляция? Что такое метод? Что такое конструктор?

Что такое принцип сокрытия?

Что такое «чёрный ящик»?

Что такое оператор?

Что такое поле класса?

Задание 3. 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>

Вопросы

Какие бывают виды конструкторов?

Что такое оператор?

Объясните правило большой пятёрки.

Что такое оператор? Как он определяется?

Чем отличается класс `vector` от класса `list`?

Задание 4. Наследование

1. Создать UML диаграмму из 3-х (или более) классов имеющих отношение типа наследование.
2. Реализовать классы на C++. Продемонстрировать работу с ними.

Вопросы

Что такое наследование?

Какие классы называется базовыми и производными?

Что такое множественное наследование?

Сколько предков может иметь класс?

Что такое перегрузка метода? Что такое переопределение метода?

Что такое множественное наследование? Чем оно опасно? Когда его можно использовать?

Что такое интерфейс (ООП)?

Задание 5. 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.

Вопросы

Какие отношения возможны между классами?

Какие отношения возможны между объектами?

Что такое мощность отношения?

Как в C++ может быть отображено отношение *ассоциация (связь)*?

Задание 6. Реализовать лабораторную 5 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.

Вопросы

Какие отношения возможны между классами?

Какие отношения возможны между объектами?

Что такое мощность отношения?

Как в C++ может быть отображено отношение *ассоциация (связь)*?

Задание ?.

Матричный калькулятор с графическим интерфейсом пользователя?

Диаграмма классов на C++ с графическим интерфейсом пользователя?

Фильтрация ввода некорректных данных. Удобный UI.

Семестр II

Задание. Виджеты Qt. Сигналы и слоты.

Продemonстрировать механизм сигналов и слотов на примере виджетов Qt. Создать как минимум один собственный виджет на основе имеющихся в Qt.

см. пример в лекции 1.

Задание. Калькулятор.

Калькулятор должен корректно обрабатывать любые входные данные. Сделать обработку исключительных ситуаций. Хранить историю вычислений. Помимо арифметических операций и возведения в любую степень калькулятор должен вычислять функции: \sin , \cos , \tan , \ln , \exp .

Задание. Матричный калькулятор.

Программа с графическим интерфейсом пользователя для операций над матрицами.

Реализовать операции над матрицами: Сложение, вычитание, умножение матриц. Умножение матрицы на число. Транспонирование. Вычисление определителя. Размерность матрицы задётся пользователем.

Задание. Игра

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