ООП. Часть 2 В этом семестре можно выбирать язык программирования из: C++, Java, C#. Можно предложить другой язык программирования, например Kotlin. Согласуйте это с преподавателем.
Вместо всех заданий кроме первого можно работать над своим проектом по согласованию с преподавателем.

Задание 1. Простой класс на Java или C#

1. Согласуйте тему с преподавателем.

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. Операторы сложения, вычитания, умножения (на комплексное и действительное число). Вычисление аргумента и модуля.
- Кватернион. Аналогично комплексному числу.
- Вектор. Задаётся своими компонентами. Вычисление длины, углов между осями; операторы сложения и вычитания, умножения на число.
- Геометрическая фигура. Задание сторон, координат на плоскости. Вычисление площади и периметра.
- CSV файл.
- Другой класс по согласованию с преподавателем. Класс может взаимодействовать с сетью, ОС, файлами и т.д.

2. Создайте класс на C# или Java. Реализуйте методы для доступа и изменения данных, конструктор с параметрами. Операторы и генерирование исключительных ситуаций если необходимо.

3. Приведите документацию к классу в машинно-читаемом формате. Опишите назначение класса, принципы использования, смысл поелй, методов и их параметров. Если необходимо привести пример использования класса в документации.

4. Создайте автоматический модульный тест. Приветствуется использование встроенных средств среды разработки для модульного тестирования.

5. Продемонстрируйте работу с классом в приложении с графическим интерфейсом. Программа не обязательна должна взаимодействовать с пользователем, главная цель — показать пример использования класса.

6. Создайте UML диаграмму для всего приложения. Приведите только названия классов.

7. *Дополнительно*: создать библиотеку с описанным классом. Продемонстрировать пример использования библиотеки.

Вопросы

1. Что такое АДТ?
2. Что такое предусловия?Для чего нужны? Что такое постусловия? Что такое

класс? Что такое объект?

3. Что такое абстрагирование?
4. Что такое инкапсуляция? Что такое метод, конструктор, оператор?
5. Что такое принцип сокрытия? Что такое «чёрный ящик»?
6. Что такое поле класса? Что такое свойство (C#)?
7. Как вызвать метод конкретного объекта находящегося в массиве? Чем отличаются обращения к методам в C++ с использованием объекта, ссылки на объект и указателя на объект?
8. Что такое равенство объектов? Когда объекты идентичны?
9. Что такое поведение? Что такое состояние?

Ссылки

- github.com/VetrovSV/OOP/blob/master/OOP_2.pdf — коротко о классах в C# и Java
- plantuml.com/ru — создание диаграмм из plain text

Задание 2. Чат-бот.

Создайте программу с графическим интерфейсом пользователя. Реализуйте логику в отдельном модуле.

Требования

- При запуске появляется окно для задания имени пользователя, потом основное окно.
- Ответ ботом на несколько реплик заданного шаблона («Привет, Бот!» и т.п.)
- Ответ на простые команды (вопросы). Например: «Который час?», вопросы о статистике по обмену сообщениями и т.п.
- Ответ на команды с параметрами: Например: «умножь 12 на 157»
- Бот должен хранить историю сообщений, включая время отправки и автора.
- Записывать историю в файл при завершении программы. Загружать из файла при запуске программы.
- *Дополнительно реализовать один или несколько пунктов (макс. Оценка на экзамене 3, если не выполнено):*
  - *Получение актуальной информации из интернета (погода, курсы валют, последние новости и т. п.)*
  - *Запуск отдельных программ, работа с операционной системой и файлами.*
  - *Сохранение информации о собеседнике. Собеседник предполагается неизменным*
  - *Опционально: показ изображений (в том числе загрузка из интернета, например APOD))*
- Требования к GUI:
  - шрифт и цветовая палитра (опционально: использование фоновых изображений) отличные от задаваемых по умолчанию.
  - иконка приложения
  - отправка сообщения по горячей клавише

Альтернативное задание: текстовый квест с аналогичными требованиями.

Рекомендации

1. Используйте библиотеки для обработки естественного языка, регулярных выражений.
2. Помните, что класс (или классы) отвечающие за обработку сообщений должны быть независимы от интерфейса программы. Их, например, можно использовать для реализации онлайн бота.
3. Старайтесь сделать эти классы гибкими. В них должно просто добавлять новые методы обработки сообщений или реакции на сообщения.
4. Используйте абстрактные классы для описания программного интерфейса.

Вопросы

1. Изобразите диаграмму классов для приложения.
2. Имеется ли интерфейс определяющий способы взаимодействия с классом обрабатывающим сообщения пользователя?
3. Что такое бизнес-логика?
4. Опишите шаблон проектирования «Модель – представление».
5. Что такое SOLID? Опишите каждый принцип.
6. Ваша лабораторная соблюдает принципы SOLID?
7. Что такое регулярное выражение?

Ссылки

- github.com/VetrovSV/OOP/tree/master/2021-fall/chat_bot – шаблон для программы на C++
- многооконные приложения в Qt:
  - github.com/VetrovSV/OOP/blob/master/Qt_lec4.pdf
  - github.com/VetrovSV/OOP/tree/master/examples/Qt/multy-window
- github.com/VetrovSV/OOP/blob/master/examples/cpp_map/main.cpp – тип данных словарь (C++)
- stackoverflow.com/questions/46943134/how-do-i-write-a-qt-http-get- request
- Фридл Дж. Регулярные выражения, 3-е издание. – Пер. с англ. – СПб.: СимволПлюс, 2008. – 608 с., ил.
- regex101.com – сайт для проверки регулярных выражений

Задание 3. Простая БД

Простая файловая БД с GUI. Требования[4]:

- Разделение представления и модели (данных и методов работы с ними).
- Одна таблица с 4+ полями.
- Собственный формат БД или SQL (QSLite или серверная БД)
- Добавление, *проверка*, изменение, удаление данных
- Поиск, *сортировка (как минимум по одному полю)*.
- Документация (в коде) описывающая формат данных в файле.
- Требования к GUI:
  - вывод данных в таблицу
  - меню приложения
  - *панель инструментов*[1]
  - *шрифт и цветовая палитра отличные от задаваемых по умолчанию.*
  - *иконка приложения*
  - Всплывающая подсказка или подсказка в строке состояния для элементов интерфейса.
  - Информация о разработчике.
  - *Дополнительно (выполните минимум 3 пункта):*
    - *горячие клавиши*
    - *цветовое кодирование данных в таблице*
    - *использование элементов интерфейса (флажок, числовое поле ввода и тт.п.) в таблице*
    - *хранение изображений в БД*
    - *использовать как минимум одно модальное окно*
    - *Автоматическое сохранение БД через заданные интервалы времени*
    - *краткая справка*

1 Курсивом отмечены необязательные необязательные требования. Максимальная оценка на экзамене — 3 если не выполнены.

Вопросы

1. Что такое представление и модель?
2. Как представлена модель в программе? Как происходит проверка данных?
3. Какие исключительные ситуации могут возникнуть во время работы программы?
4. Что такое SOLID? Соблюдаются ли эти принципы в вашей программе?

Ссылки

- Считывание данных из файла CSV и их представление через QstandardItemModel evileg.com/ru/post/158
- raw.githubusercontent.com/VetrovSV/OOP/master/Qt_lec2.pdf
- Пример на Qt: github.com/VetrovSV/OOP/tree/master/examples/Qt/database
- DataGrid и BindingList в C#: github.com/VetrovSV/OOP/tree/master/examples/WpfApp_DataGrid
- Qt StyleSheet
  - doc.qt.io/qt-5/stylesheet-examples.html
  - doc.qt.io/qt-5/stylesheet-reference.html
  - Пример: github.com/VetrovSV/OOP/blob/master/examples/Qt/database/style.qss

Задание 4. Виджеты Qt. Сигналы и слоты

Не обязательна. Если не выполнена максимальная оценка на экзамене — 4

Продемонстрировать механизм сигналов и слотов на примере виджетов Qt.

Например соединить 2-3 метода одного виджета с другим. Возможно использование лямбда функций.

Вопросы

1. Что такое сигнал и слот? Как они работают?
2. Как соединить сигнал со слотом?
3. Сколько сигналов могут быть соединены со слотом? Слотов с сигналом? 4.Какие есть требования к классу для использования его методов как сигналов и слотов?
4. Как передавать данные с помощью сигналов и слотов? 6.Как соединить сигнал с лямбда-функцией?
5. Опишите объектную иерархию в Qt
6. Как происходит компиляция проекта использующего Qt?

Задание 5. Игра

(Не обязательно. Если выполнено +1 к оценке на экзамене)

Любой объектно-ориентированный язык общего назначения на выбор.

- Возможные варианты: Игра «Жизнь» Конвея, Сапёр, Тетрис, Арканойд, Морской бой;
- или свой вариант (по согласованию с преподавателем);
- или программа для моделирования роботов на двумерной плоскости.
  - *Схематичное отображение.*
  - *Робот может перемещаться на клетку, поворачивается, атаковать или собирать ресурсы лежащие на плоскости.*
  - *Поведение робота определяет класс с заданной спецификацией - интерфейс. Все конкретные реализации робота — его потомки.*
  - *Дополнительно: выполнение каждого класса в отдельном потоке.*

Вопросы

1. Изобразите диаграмму классов для приложения
2. Что такое регулярное выражение?
3. Что такое SOLID? Опишите каждый принцип.
4. Ваш код не нарушает принципов SOLID?

Задание 6. UI markup language

(Не обязательно. Если не выполнено макс. оценка на экзамене - 4)

Простое приложение с GUI построенным с использованием языка разметки UI (QML, XAML, FXML и др). Пример приложения – вычисление дохода по вкладу с использованием сложных процентов.

Вопросы

1 Что такое декларативный язык?
2 Что такое UI markup language? Приведите примеры.
3 Какие ещё существуют способы дизайна графического пользовательского интерфейса?
4 Преимущества использования языка описания UI?

Задание 7. Приложение готовое к развёртыванию

(Не обязательно. Если не выполнено макс. оценка на экзамене - 4)

Создать установочный файл (для Windows, MacOS или Linux) для любого из) для любого ранее созданных приложений.

Для создания инсталлятора использовать готовые инструменты.

Дополнительно: использовать цифровую подпись.

Задание 8. Приложение для Andoid.

(Не обязательно. *Если выполнено +1 к оценке на экзамене*)

Скомпилировать одно из ранее созданных приложений для Android версии 9+.