

## Семестр II

### Задание 1. Простой класс на Java или C#<sup>3</sup>

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. Здаётся своими компонентами. Вычисление длины, углов между осями; операторы сложения и вычитания, умножения на число.
- Другой класс по согласованию с преподавателем.

1. Представить класс в виде UML диаграммы

2. Описать класс на C# или Java. Реализовать методы для доступа и изменения данных, конструктор с параметрами. Операторы и генерирование исключительных ситуаций если необходимо. Продемонстрировать работу с классом в приложении (GUI). Программа не обязательно должна взаимодействовать с пользователем, главная цель — показать пример использования класса.

3. Для класса привести документацию описав его назначение, принципы использования, смысл методов и их параметров. Если необходимо привести пример использования класса в документации.

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

---

<sup>3</sup> Можно предложить другой язык программирования, например Kotlin.

### Вопросы

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

## Задание 2. 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?
5. Как передавать данные с помощью сигналов и слотов?
6. Как соединить сигнал с лямбда-функцией?
7. Опишите объектную иерархию в Qt
8. Как происходит компиляция проекта использующего Qt?

### Задание 3. Простой чат бот.

Программа с GUI. Любой объектно-ориентированный язык общего назначения на выбор. Бизнес-логика в отдельном модуле. Требования:

- Сначала появляется отдельное окно для задания имени пользователя (авторизации), 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

Альтернативное задание: текстовый квест с аналогичными требованиями.

Рекомендуется использование библиотек для обработки естественного языка, регулярных выражений.

**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**

<https://stackoverflow.com/questions/46943134/how-do-i-write-a-qt-http-get-request>

## Задание 4. Простая БД

Простая файловая БД с GUI. Любой объектно-ориентированный язык общего назначения на выбор. Требования<sup>4</sup>:

- Разделение представления и модели (данных и методов работы с ними).
- Одна таблица с 4+ полями.
- Собственный формат БД (без использования SQL, noSQL и проч.).
- 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*
  - *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:*
    - *горячие клавиши*
    - *цветовое кодирование данных в таблице*
    - *использование элементов интерфейса (флажок, числовое поле ввода и т.п.) в таблице*
    - *хранение изображений в БД*
    - *использовать как минимум одно модальное окно*
    - *Автоматическое сохранение БД через заданные интервалы времени*
    - *краткая справка*

---

<sup>4</sup> Курсивом отмечены 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 <https://evileg.com/ru/post/158/>

## 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 ранее созданных приложений.

Для создания инсталлятора использовать готовые инструменты.

*Дополнительно: использовать цифровую подпись.*

## Задание 8. Юнит-тест

(Не обязательно. Если не выполнено макс. оценка на экзамене - 3)

Набор юнит тестов для ранее созданного класса (классов)

### Вопросы

1. Что такое юнит-тест?
2. Для чего нужны юнит-тесты? В чём их преимущество?
3. Что такое Test Driven Development?
4. Приведите пример когда создание юнит-тест оправдано и когда нет.

Пример: [https://github.com/VetrovSV/OOP/tree/master/examples/simple\\_class\\_tests](https://github.com/VetrovSV/OOP/tree/master/examples/simple_class_tests)



**Задание 9. Приложение для Andoid.**

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

Скомпилировать одно из ранее созданных приложений для Android версии 6+.