Лабораторные работы № 3-4
Объектно-ориентированное проектирование системы. Язык UML

Цели работы:
- изучить процесс объектно-ориентированного проектирования системы – от анализа требований до генерации кода
- научиться строить основные диаграммы UML
- получить навык использования CASE-средств проектирования

Инструментарий:
- Git (исходные данные в вашем репозитории)
- Средство построения диаграмм UML (например, Rational Rose, Visual Paradigm, Magic Draw)
- Текстовый редактор

Теоретические сведения:
1. Лекционный материал по языку UML и объектно-ориентированному проектированию (+ лекции из КПП – например, тема «Абстрактные структуры данных»)
2. Методические материалы "Практикум ООПиП"

Порядок выполнения работы:
1. ~~Изучить теоретические сведения.~~
2. Построить модель предметной области в виде диаграммы классов (используя спецификацию требований), составить глоссарий для ключевых понятий.
3. На основании раздела “Системные требования” спроектировать Use Case представление системы (источником информации об актёрах может быть также подраздел «Характеристики пользователей»).
4. Для каждого варианта использования описать сценарий как можно более подробно (учитывая элементы GUI, обработку неправильного ввода, ошибок, альтернативные действия, краткое описание задействованных актёров). Оформить сценарии в текстовом виде как описание “Потока событий” (Flow of events) – формат и примеры в «Практикуме»
5. ~~Представить ключевые сценарии в виде диаграмм активностей (activity)~~
6. На основании диаграмм активностей, используя и уточняя объекты из модели предметной области, построить диаграммы последовательности.
7. ~~Построить диаграммы состояний (например, для состояний GUI). Использовать мокапы.~~
8. Уточнить диаграмму классов с учётом построенных ранее в работе диаграмм (методы диаграмм последовательности, структура объектов, поля, состояния и т.д.)
9. ~~Построить диаграмму компонентов (показать размещение объектов на компонентах или других артефактах, отобразить артефакты на компоненты)~~

10. ~~Построить диаграмму развёртывания, показать отображение программных компонентов на реальные физические узлы.~~
11. ~~Все результаты хранятся в репозитории.~~
12. Рекомендуется выполнять работу поэтапно:
    a. Use Case (с описанием сценариев), Activity, Sequence
    b. State, Class, Component/Deployment
13. При использовании Case-средств обязательно придерживаться структуры проекта этого средства (В Rational Rose: Меню Tools-> Check Model - проверить, есть ли в проекте ошибки)