**ЛАБОРАТОРНАЯ РАБОТА №2**

**«Проектирование программной системы с использованием CRC-карт»**

**Цель работы.** Изучить метод проектирования систем посредством CRC-карт

**Понятия и определения, используемые в работе**

**Актор** – множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями.

**Прецедент** – возможность моделируемой системы (то, что система делает, часть функциональности системы, необходимая для получения результаты пользователем этой системы).

**Объектно-ориентированное программирование** – методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

**Класс** – это элемент ПО, описывающий абстрактный тип данных и его частичную или полную реализацию.

**Подкласс** – это класс, наследующий некоторые (или все) свойства от своего суперкласса.

**Суперкласс** – класс, на основе которого создаются другие классы.

**Теоретические сведения**

Одним из важнейших этапов создания программного продукта является стадия проектирования системы. На данном этапе закладывается структура связей отдельных элементов – создается архитектура системы.

Используя концепцию объектно-ориентированного программирования при создании программного обеспечения возможно использовать метод CRR-карт (Class-responsibility-collaboration card, карта «Класс – ответственность – кооперация»). Концепция использования CRC-карт была предложена Уордом Каннингемом и Кентом Беком.



Рис. 2.1 – Иллюстрация CRC-карты (из оригинальной работы Уорда Каннингена и Кента Бека [1])



Рис. 2.2 – Иллюстрация использования CRC-карт для описания MVC модели Smalltalk (из оригинальной работы Уорда Каннингена и Кента Бека [1])

CRC-карты применяются в том случае, когда изначально необходимо определиться с классами и способами их взаимодействия на стадии проектирования программного обеспечения.

Типичное содержание CRC-карты:

1. Класс (подкласс, суперкласс)

2. Обязанности класса

3. Связь с другими классами

Применение малых карточек позволяет абстрагироваться от сложности разработки. Карты акцентируют внимание на сущности класса и скрывают от разработчика детали, которые не нужны на стадии проектирования. Малый размер карточек на психологическом уровне способствует выделению главных обязанностей класса.

В процессе проектирования с использованием CRC-карты выделяются следующие этапы:

**1. Построение диаграммы прецедентов.** На данном этапе выявляются акторы и прецеденты системы.

Рассмотрим диаграмму прецедентов на примере банкомата.



Рис. 2.3 – Диаграмма прецедентов на примере системы «Банкомат»

Прецедентов может быть огромное множество. На примере системы «Банкомат» можно добавить также такие варианты использования как: проверить пароль, вывести информацию на экран и т.д.

Данный вид диаграмм применяется также в системе управления требованиями. Разработчик, используя диаграмму может показать основной функционал программного обеспечения, планируемого к разработке и убедиться в правильности необходимых реакций на отдельный действия.

**2. Определение классов.** На данном этапе определяются классы необходимые для разработки системы. На примере банкомата это: клиент, банк, банкомат, служба безопасности банка и др. Классы описываются посредством CRC-карт. Физически карты часто представляют собой листки бумаги размером 10\*15 см и разделенные на 3 части.

|  |
| --- |
| Название класса |
| Действия, которые он выполняет (принято обозначать с глагола) | Классы, с которыми происходит взаимодействие |

В случае системы «банкомат» карты будут иметь следующий вид:

|  |
| --- |
| Клиент |
| Вставляет карточку в банкоматВводит парольУказывает тип операцииВводит суммуПолучает деньгиВынимает карточку | Банкомат |

|  |
| --- |
| Банкомат |
| Отображает информацию для клиентаПередает информацию в банкВыдает купюрыРаспечатывает счет | КлиентБанкСлужба безопасности банка |

|  |
| --- |
| Служба безопасности банка |
| Проверяет парольПроверяет подлинность карточкиИдентифицирует клиентаСледит за правильностью транзакций | БанкБанкомат |

|  |
| --- |
| Банк |
| Проверяет возможность выдачи средствСообщает о наличии денегВыдает информацию об остаткеХранит информацию о счете клиента | БанкоматСлужба безопасности банка |

**3. Проверка достаточности (избыточности) классов и корректности их взаимодействия.** На данном этапе строится диаграмма взаимодействия спроектированных классов.



Рис. 2.4 – Диаграмма взаимодействия

На основании данного метода можно выполнить инсценировку работы системы. Для этого достаточно будет раздать участникам разработки карточки с классами. После чего начать импровизированную ролевую игру, в которой первый участник читает действие исходе из доставшейся ему карточки с классом, а другие участники соответствующим методом реагируют на его действия. Такой метод может определить как избыточные взаимодействия, так и явные ошибки про проектировании.

**Задание**

Разработать проект системы по методу CRC-карт для одного из следующих вариантов.

**Варианты задания:**

1. Заказ билетов в аэропорту.
2. Электронный магазин.
3. Отправка sms.
4. Пропускная система.
5. Компьютерная система тестирования для оценки знаний студентов.
6. Заказ билетов в театральной кассе.
7. Телефонный коммутатор.
8. Система учета успеваемости студентов деканатом.
9. Записная книжка.
10. Пропускная система биоконтроля.
11. Автомат платежной системы.
12. Электронная почта.

**Требования к содержательной части отчета**

В отчете необходимо представить:

1. Цель работы

2. Номер варианта и наименование информационной системы для разработки.

3. Схематическую диаграмму прецедентов (нет надобности использовать в полной мере правильную нотацию согласно требованиям стандарта UML)

4. CRC-карты (выполняются в виде таблиц, по аналогии с примерами)

5. Диаграмму действий (нет надобности использовать в полной мере правильную нотацию согласно требованиям стандарта UML)

6. Текст описания диаграммы взаимодействия.

**Библиографический список:**

1. ***K. Beck, W. Cunningham*** A Laboratory For Teaching Object-Oriented Thinking OOPSLA '89 Conference proceedings on Object-oriented programming systems, languages and applications URL: http://www.inf.ufpr.br/andrey/ci221/docs/beckCunningham89.pdf