**АНАЛИЗ ТРЕБОВАНИЙ**

Одни из методик определения обьектов — проанализировать требования. Например:

**user story**

nouns

Я, как **онлайн-покупатель**, хочу добавить **товар** в **корзину**, что бы я мог купить его

Имена существительные — будут 3 обьекта системы

verbs

Я, как онлайн-покупатель, хочу **добавить** товар в корзину, что бы я мог **купить** его

Глаголы в данном случае отвечают операциям корзины

Так же требования могут помочь определить отношения между обьектами, в данном случае обычно 1 покупатель ассоциируется с 1 корзиной, корзина может содержать множество единиц товара

**Категории обьектов проекта**

**Обьекты сущностей** —отвечают обьектам реального мира, например юзер, или дом

**Обьект контроля** — обьект, который координирует другие обьекты, в тоже время оставляя их слабо связанными. Отичный паттерн — Mediator

**Обьект границы** — граничат между системами. Например обьект, который запрашивает информацию в нете; или обьект, который выводит информацию для юзера или забирает его инпут;

**Концептуальное проектирование** определить основные обьекты задачи

**Техническое проектирование** (более углубленный, после концептуального) — определяет детали обьектов, включая их свойства и поведение

**Основные принципы проектирования:**

- гибкость

- реюзабельность

- поддерживаемость

**Концептуальное проектирование — CRC**

**CRC** — карточка Class, Responsibility, Collaborators

Class — сам класс, пишется вверху каты

Resposibility — обязанности класса, пишутся под классом справа

Collaborators — другие классы, с которыми класс взаимодействует что бы выполнять свои обязанности

Пример: банковский автомат. Приходит юзер, вставляет карточку, автомат просит ввести ПИН-код, после чего юзер может на выбор: снять деньги, посмотреть баланс или …

|  |  |
| --- | --- |
| Пользователь | |
| Resposibilities  - вставить карточку  - выбрать операцию | Collaborators  - Банковский автомат |

|  |  |
| --- | --- |
| Банковский автомат | |
| Resposibilities  - аутентифицировать юзера  - отобразить опции выбора  - снятие и запихивание наличных  - проверка баланса | Collaborators  - Пользователь |

**Абстрактный тип данных**

Тип данных, который не встроен в язык и определяется програмистом. Это группировка связанной информации

**Цель обьектно-ориентированного программирования:**

* Сделать абстрактный тип данных более легким к написанию
* Построить систему вокруг абстрактных типов данных, называемых классами
* Представить возможность для абстрактных типов данных расширять друг друга

**ЧЕТЫРЕ ПРИНЦИПА ПРОЕКТИРОВАНИЯ**

**Абстракция** — идея упрощения концепта в проблемной области. Она разбивает задачу на упрощенное описание , которое игнорирует несущественные детали и подчеркивает основные, необходимые для концепции

**Rule of least astonishment.** Абстракция должна следовать правилу наименьшего удивления. Это правило предполагает, что основные атрибуты и поведение должны быть описаны без сюрпризов и без каких-либо определений, выходящих за рамки необходимых. Это отсекает нерелевантные характеристики как часть абстракции и помогает убедиться, что абстракция имеет смысл для целей концепции.

Абстракция содержит **атрибуты** и **поведение:**

Абстракция собаки, атрибуты:

- имя

- порода

- возраст

- цвет

Абстракция собаки, поведение:

- есть

- спать

- бегать

- прыгать