

Лекция 1.2. Класове и обекти

Когато използваме обектно ориентирани методи за анализ и дизайн на сложна софтуерна система, нашите основни градивни блокове са класовете и обектите.

- Същност на обекта

Способността да разпознават физически обекти е умение, което хората придобиват в много ранна възраст.

Неформално обектът е осезаема същност (*entity*), която проявява някакво добре дефинирано поведение. От гледна точка на човешкото познание обект е всяко от следните:

- нещо осезаемо и/или видимо;
- нещо, което може да се възприеме интелектуално;
- нещо, към което е насочена мисъл или действие.

Към неформалната дефиниция добавяме идеята, че обектът моделира някаква част от реалността и за това е нещо, което съществува във времето и пространството. В софтуера терминът „обект“ се използва за първи път в езика Simula. Обектите в Simula обикновено симулират някои аспекти на реалността.

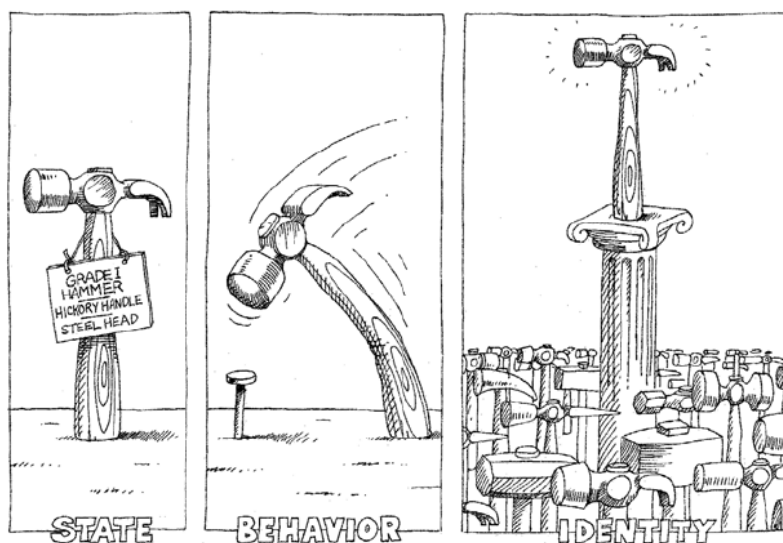
Някои обекти могат да имат отчетливи концептуални граници и въпреки това да представят нематериални събития или процеси. Например, химически процес в производствено предприятие може да се третира като обект, защото има ясна концептуална граница, взаимодейства с други обекти чрез набор от операции и проявява добре дефинирано поведение.

Какво не е обект?

Някои предмети могат да бъдат осезаеми и въпреки това да имат размити физически граници, например реки, мъгла, тълпа от хора. Не всичко е подходящо за обект. Например, атрибути като красота или цвят не са обекти. Същото се отнася за емоции като любов и гняв. От друга страна, тези неща са потенциални характеристики (*properties*) на други обекти. Можем да кажем, например, че мъжът (обект) обича своята съпруга (друг обект), или че дадена котка (още един обект) е сива.

Като изхожда от натрупания опит, Grady Booch предлага следната дефиниция:

Дефиниция: *Обект е същност, която има състояние, поведение и идентичност. Структурата и поведението на сходни обекти са дефинирани в техния общ клас. Термините „инстанция“ и „обект“ са взаимнозаменяеми.*



Фиг. 1 Състояние, поведение и идентичност на обекта

Състоянието (*state*) на обекта включва всички негови (обикновено статични) характеристики плюс текущите (обикновено динамични) стойности на всяка от тези характеристики.

Характеристика (*property*) е присъщо или отличително свойство, черта качество или функция, която допринася за еднозначното идентифициране на обекта. Всички характеристики имат някаква стойност. Тя може да бъде просто количество или може да означава друг обект.

Фактът, че всеки обект има състояние предполага, че всеки обект заема някакво пространството, било то във физическия свят или в паметта на компютъра.

Поведение

Никой обект не съществува в изолация. Обектите действат върху други обекти, както и други обекти действат върху тях. Можем да кажем, че:

Дефиниция: Поведението (behavior) е как обекта действа и реагира чрез промени на състоянието си и предаване на съобщения.

С други думи, поведението на обекта представлява неговата видима активност.

Операция е действие, което един обект извършва върху друг, за да предизвика реакция. Например клиент може да извика операция `Add()`, за да добави нов елемент към `List<string>`. Клиентът също може да извика операция `Count` (реализирана като *property*), която връща стойност, означаваща броят елементи в списъка, но не променя неговото състояние.

Операциите, които клиентите могат да извършват с обект обикновено се дефинират като член-функции (методи). Извикването на метод на даден обект в чистите обектно ориентирани езици се нарича „изпращане на съобщение“.

Изпращането на съобщения е само едната страна на определянето на поведението на обекта. Неговото състояние също може да влияе на поведението, например автомат за напитки ще реагира по различен начин на едно и също съобщение, в зависимост от това дали пуснатата сума е достатъчна.

Може да се каже, че поведението на обекта е функция на неговото състояние и операцията, извършвана върху него, като отчитаме факта, че някои операции имат като страничен ефект промяна на състоянието на обекта.

Тогава можем да уточним нашата дефиниция за „състояние“:

Състоянието на обекта представлява кумулативен резултат от неговото поведение.

Операции

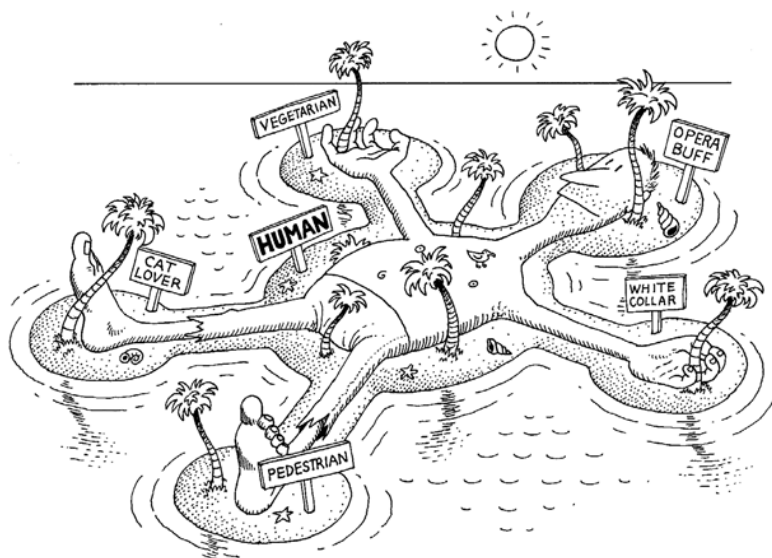
Операцията обозначава услуга, която класът предоставя на своите клиенти. В практиката са открити 5 типични операции, които клиентите извършват върху обекти:

- модификатор: операция, която променя състоянието на обекта;
- селектор: операция, която осигурява достъп до състоянието на обекта, но не го променя;
- итератор: операция, която позволява да се обхождат всички части на обекта в строго определен ред;
- конструктор: операция, която създава обект и/или инициализира неговото състояние;
- деструктор: операция, която освобождава състоянието на обекта и/или разрушава самия обект.

Роли и отговорности

Всички методи, свързани с даден обект конституират неговия **протокол**. Протоколът на обекта по този начин определя обвивката от допустимото за него поведение и така обхваща целия статичен и динамичен изглед на обекта. За повечето нетривиални абстракции е полезно протоколът да се раздели на логически групи на поведение. Тези групи обозначават ролите, които обектът

може да изпълнява. **Ролята** е маска, която обектът носи и така определя договор между абстракцията и нейните клиенти.



Фиг. 2 Един обект може да играе много различни роли

Отговорностите (*responsibilities*) имат за цел да създадат усещане за целта на обекта и неговото място в системата. Отговорностите на даден обект са всички услуги, които той предоставя за всички договори, които поддържа. С други думи, състоянието и поведението на обекта взети заедно определят ролите, които този обект може да изпълнява. Тези роли от своя страна изпълняват отговорностите на абстракцията.

Идентичност

Дефиниция: *Идентичност (identity)* е свойството на обекта, което го разграничава от всички останали обекти.

Трябва да се отбележи, че в повечето езици за програмиране и БД се използват имена на променливи за разграничаване на временни обекти, като по този начин се смесват адресируемостта и идентичността. Повечето СУБД използват ключове за разграничаването на обекти, като по този начин се смесват стойност и идентичност. Неосъзнаването на разликата между името на обекта и самия обект е източник на много грешки в ООП.

Един обект може да има няколко имена, както и едно име може да се свързва с различни обекти.

Отношения между обектите

Обектите допринасят за поведението на системата като си взаимодействат. Отношението между всеки два обекта обхваща предположенията, които всеки прави за другия, включително какви операции могат да се извършват и какво поведение ще се получи в резултат. За ООАД интерес представляват два вида отношения между обектите: връзки (*links*) и агрегация (*aggregation*).

Връзки

Обектът си сътрудничи с други обекти чрез своите връзки с тях. Връзките обозначават специфична асоциация, чрез която един обект (клиентът) прилага услугите на друг обект (доставчик) или чрез която един обект може да навигира към друг.

Като участник във връзка, обектът може да изпълнява една от следните 3 роли:

- контролер (*controller*): този обект може да оперира с други обекти, но с него никога не оперира;
- сървър: с такъв обект само се оперира, без той да оперира с други обекти;

- **пълномощник (proxy):** този обект може да оперира с други обекти, както и други обекти могат да оперират с този обект. Обикновено прокситата представят абстракция на реален обект в предметната област на приложението.

Агрегация

Докато връзките обозначават равноправни (*peer-to-peer*) или клиент/сървър отношения, агрегацията означава йерархия от вида цяло/част (*whole/part*) с възможност за навигиране от цялото (наричано още агрегат) към неговите части. В този смисъл агрегацията е специален вид асоциация.

Агрегацията може да не означава физическо съдържане, например връзката между акционер и неговите акции, е такова отношение на агрегация – той притежава акциите, но те не са физическа част от него.

Често се налага да се избира между връзки и агрегация. Агрегацията понякога е по-доброто решение, защото капсулира частите като тайни на цялото. Връзките също понякога са по-добрата алтернатива, защото те позволяват по-свободно свързване между обекти. Интелигентните инженерни решения изискват внимателно претегляне на тези два фактора.

- Същност на понятието „клас“

Понятията „клас“ и „обект“ са плътно преплетени, така че не можем да говорим за обект без да вземем предвид неговия клас. Въпреки това, съществуват важни различия между тези две понятия.

Докато обектът е конкретна същност, налична във времето и пространството, класът представя само абстракция на същественото за обекта. Например, можем да говорим за клас „Бозайник“, който представя общите за всички бозайници характеристики. За да идентифицираме отделен бозайник от този клас трябва да кажем „този бозайник“ или „онзи бозайник“.

В контекста на ООАД можем да дефинираме класа така:

Дефиниция: *Клас е множество от обекти, които споделят обща структура, общо поведение и обща семантика.*

Отделният обект е просто инстанция на класа.

Интерфейсът на класа предоставя неговия външен изглед и за това акцентира на абстракцията, скривайки неговата структура и тайните от неговото поведение. Този интерфейс се състои главно от декларации на всички операции, приложими към обекти на класа, но може да съдържа също декларации на други класове, константи, променливи и изрази, необходими за завършения вид на абстракцията.

За разлика от интерфейса, **имплементацията** на класа е неговия вътрешен изглед, който обхваща тайните на неговото поведение. Имплементацията на класа се състои главно от имплементациите на неговите методи (операциите), деклариран в интерфейса на класа.

Отношения между класовете

Във всяка отделна проблемна област ключовите абстракции обикновено са свързани по различни интересни начини, формирайки структурата на класовете в дизайна.

Има **три основни вида отношения между класовете**:

- **обобщение/специализация** (*generalization/specialization*): обозначава „is a“ релация. Например, „розата е вид цвете“ означава, че роза е специализиран подклас на по-общия клас цвете;
- **цяло/част** (*whole/part*): обозначава „part of“ релация. Например, венчелистче не е вид цвете, то е част от цвете;

- **асоциация** (*association*): означава някаква семантична зависимост между иначе несвързани класове, като между калинките и цветята. Друг пример – рози и свещи са доста независими класове, но и двата представят неща, които можем да използваме за декорация на масата за вечеря.

Ролята на обектите и класовете в анализа и дизайна

По време на анализа и ранните етапи от дизайна на софтуерната система разработчикът има две основни задачи:

1. Да открие класовете, които формират речника на проблемната област.
2. Да измисли структури, при които множества от обекти работят съвместно за предоставяне на поведения, които удовлетворяват изискванията на проблема.

Наричаме тези класове и обекти с общото име **ключови абстракции** на проблема, а коопериращите се структури – **механизми** на имплементацията.

По време на тези фази от разработката трябва да се обмисли външният изглед на ключовите абстракции и механизми. Този изглед ще представя логическата рамка на системата и за това обхваща структурата на класовете и структурата на обектите в системата. В по-късните етапи от дизайна и прехода към имплементирането, задачата на разработчика се променя: фокусът е върху вътрешния изглед на ключовите абстракции и механизми, замесвайки тяхното физическо представяне.

Обобщение

- обектът има състояние, поведение и идентичност;
- структурата и поведението на сходни обекти се дефинира в техния общ клас;
- състоянието на обекта обхваща всичките му (обикновено статични) характеристики плюс текущите (обикновено динамични) стойности на всяка от тези характеристики;
- поведението е как обектът действа и взаимодейства в термините на промяна на неговото състояние и предаване на съобщения;
- идентичността е свойство на обекта, което го отличава от всички останали обекти;
- класът е множество от обекти, които споделят обща структура и поведение;
- трите вида отношения са асоциация, наследяване и агрегация;
- ключовите абстракции са класове и обекти, които формират речника на проблемната област;
- Механизмът е структура, посредством която множество от обекти работи съвместно за предоставяне на желано поведение.