**1 слайд.**

Начнем с определения **Агентно-ориентированной парадигмы программирования.**

**Агентно-ориентированный подход** (в дальнейшем АОП) к программированию — разновидность представления программ или парадигма программирования, в которой основополагающими концепциями являются понятия **агента** и его **поведение**, зависящее от **среды**, в которой он находится.

В данном определении выделены основные понятия АОП подхода, которые стоит рассмотреть подробнее.

**2 слайд.**

**Программный агент** — это программа, которая вступает в отношение посредничества с пользователем или другой программой.

Самое важное, что стоит сказать об агенте, так это то, что он обладает **поведением**: то есть взаимодействует с внешней средой, способной модифицироваться или быть модифицированной другими агентами в зависимости от конкретных условий.

**3 слайд.**

Разные авторы по-разному характеризуют агента, но, в целом, в большей части определений, агенты обладают следующими свойствами:

* **Живучесть** - код не исполняется по требованию, а работает непрерывно и решает сам, когда он должен совершить какие-либо действия;
* **Автономность** - агент имеет способность выбора задач, приоритетов, целенаправленного поведения и принятия решений без участия человека;
* **Социальное поведение** - агенты имеют возможность привлекать другие компоненты посредством определённых связей и координации действий, они могут совместно работать над выполнением задач;
* **Реактивность** - агенты воспринимают контекст, в котором они работают, и реагируют на него должным образом.

**Многоагентная система (МАС)** – это система, в которой поставленная задача решается за счет совместной деятельности многих агентов в некоторой среде.

На этом этапе примерно становится ясно, чем же все-таки агенты отличаются от объектов ООП.

**4 слайд.**

Отличие агентов от объектов:

* Агенты более автономны, чем объекты;
* Агенты имеют более гибкое поведение и обладают реактивностью, проактивностью, социальным поведением.

**5 слайд.**

Приведем некоторые примеры агентов.

Вообще, агентом является всё, что может рассматриваться как воспринимающее свою среду с помощью датчиков и воздействующее на эту среду с помощью исполнительных механизмов.

Это показано на этой картинке.

Например, самые популярные агенты — это боты, будь то в играх или в торговле. Он относятся соответственно к 1 и 2 категории.

Агенты по мониторингу и наблюдению используются для наблюдения за объектами и передачи информации на оборудование, как правило, на компьютерные системы.

Например, в лаборатории реактивного движения НАСА есть агент, который отслеживает наличие и состояние оборудования, планирует заказы на приобретение нового оборудования с учётом оптимизации затрат, а также следит за наличием продуктов питания. Такие агенты обычно осуществляют мониторинг сложных компьютерных сетей и могут следить за конфигурацией каждого компьютера, подключенного к сети.

Агенты по добыче и анализу данных используют информационные технологии для поиска тенденций и закономерностей в обилии информации из различных источников. Пользователь может сортировать эту информацию для того, чтобы отобрать именно то, что он ищет.

Агенты по добыче и анализу данных работают в хранилище данных обнаруженной информации. «Хранилище данных» объединяет информацию из множества разных источников. «Анализ данных» представляет собой процесс просмотра данных в хранилище с целью нахождения такой информации, которую можно использовать для предпринятия целенаправленных действий, например, для увеличения продаж или сохранения клиентов, которые рассматривают возможность ухода к конкурентам.

**6 слайд.**

Мы более-менее разобрались с агентом и его поведением, теперь де перейдем к тому, что его поведение формирует.

**Среда** с точки зрения агента – это все, что вне его и при этом доступно для восприятия и изменения.

В зависимости от степени свободы среды, подразумевающей наличие в ней соответствующего типа агента, среды подразделяются на:

* **Замкнутые**- конечное детерминированное или вероятностное описание всей среды, которое известно агенту априори или путём исследования.
* **Открытые *-*** конечное детерминированное или вероятностное описание локальной области среды, в которой находится агент и которое известно ему априори или путём исследования.
* **Трансформируемые***-* динамически развивающиеся среды, развивающей структурой которых является агент.

**7 слайд.**

Подытожим основные концепции АОП сравнением с ООП.

Агентно-ориентированный подход является частным случаем (специализацией) объектно-ориентированного программирования (ООП).

Наличие у агента механизма целеобразования обеспечивает принципиально новый уровень автономии. Агент не обязательно выполняет распоряжения какого-либо другого агента или пользователя, он просто зависит от условий среды, включая цели и намерения других агентов. В отличие от объекта, агент может принять на себя определенные обязательства или, наоборот, отказаться от выполнения некоторой работы, мотивируя это отсутствием компетентности, занятостью другой задачей и т. п.

В то же время агент может выполнять такие действия как порождение, подавление и замена других агентов, активизация функций (как своих, так и у других агентов), активизация сценария деятельности, запоминание текущего состояния других агентов и пр.

**8 слайд.**

Перед те, как мы перейдем к конкретным примерам АОП языков, стоит разобраться с так называемой BDI-моделью агента.

Агенты рассматриваются как посредники, делегаты, искусственные представители интересов людей в некоторой виртуальной (или реальной) среде.

Поэтому, когда говорят о поведении агентов и характеризуют их состояние, то употребляют такие термины как:

* **Убеждения** (Beliefs) — это декларативные знания, которые считаются истинными с точки зрения данного агента;
* **Желания** (Desires) — это цели существования агента, т. е. его целевые состояния;
* **Намерения** (Intention) — это планы по достижению целей, принятые к исполнению.

**9 слайд.**

Конкретными реализациями данной парадигмы являются, например, следующие языки, о которых в интернете довольно мало информации:

* JADE
* Jason
* GOAL
* SARL Language

Я же нашел довольно подробное описание языка Jason, который постараюсь разобрать на следующих слайдах.

**10 слайд.**

**Jason** – это перспективное инструментальное средство для создания МАС на основе комбинации двух языков – расширенного AgentSpeak и Java, разработанное сотрудниками Австралийского технологического университета.

**AgentSpeak** – это язык описания логики работы агентов и взаимодействий между ними, ориентированный на программирование BDI-модели.

Java в Jason используется для программирования моделей специфических сред функционирования агентов, разработки пользовательских интерфейсов МАС и модификации стандартной работы самого Jason.

Фактически Jason – это интерпретатор AgentSpeak, обладающий возможностью расширения функциональности за счет открытости и использования Java.

Я не буду затрагивать Java-часть этого языка, так как она не особо нас интересует в данном контексте.

Мы же сконцентрируемся на реализации BDI-модели в этом языке.

**11 слайд.**

Убеждения в Jason формализуются в виде высказываний, предикатов или их строгих отрицаний, а также в виде правил, устанавливающих функциональную зависимость одних выражений от других.

Убеждения хранятся в базе убеждений агента, описывают декларативную составляющую его знаний, и могут меняться в процессе его работы.

**Примеры**:

* **factorial(0,1).** – убеждение-предикат, утверждающее, что факториал нуля равен единице.
* **p(X,Y):-q(X,Z)&d(Z,Y).** – предикат p(X,Y) станет убеждением, если имеются 2 убеждения q(X,Z) и d(Z,Y) с общим термом Z.

**12 слайд.**

Цели или они же намерения.

Цели подразделяются на **достигаемые** и **цели-проверки**.

**Достигаемые цели** – это желаемые убеждения, для формирования которых в обязательном порядке должен существовать план.

**Цели-проверки** – это цели, заключающиеся в определении значений термов заданных убеждений для дальнейших операций над ними в теле плана.

Примеры:

* **!print\_fact(5).** – цель - формирование убеждения print\_fact(5);
* **?inLove(bob, X).** – в базе убеждений будет найден предикат inLove(bob, X) и произойдет конкретизация переменной X.

**13 слайд.**

**Планы** описывают процедурную составляющую знаний агента.

План состоит из **четырех** частей:

* **Метки плана**(необязательна) – уникальный идентификатор, имя метки начинается с символа «@»;
* **События активации плана**-любое изменение, происходящее с убеждениями и целями, имена событий начинаются со знаков «+» и «-»;
* **Контекстных ограничений**(могут отсутствовать) - дополнительные условия активации планов;
* **Содержания плана**(может отсутствовать) — это последовательность операций, реализующих предназначение плана. Обобщенная синтаксическая конструкция плана выглядит следующим образом: **«Имя события: Контекстные ограничения <- Тело плана.»**

**14 слайд.**

Вообще, все преимущества данного подхода программирования становятся заметны только при написании крупных проектов, так как здесь даже простейшая абстракция, агент, представляет из себя довольно сложную сущность, но я все же привел пример кода на этом языке.

Попробуем его разобрать.

Здесь у нас два разных плана, которые активируются при возникновении одного и того же убеждения fact(X,Y), но в разных контекстных условиях - «X < 5» и «X == 5». В данном примере следует обратить внимание на операцию добавления нового убеждения, использованную в теле первого плана.

Эта операция обозначается символом «+» перед именем добавляемого убеждения. Ее выполнение на следующем цикле работы агента приведет к возникновению одноименного события, которое активирует один из планов. В примере показано рекурсивное выполнение планов по событиям добавления новых убеждений пока соблюдаются контекстные ограничения.

Первый план можно перевести на русский язык следующим образом: «Если возникло убеждение, что факториал X равен Y и выполняется ограничение, что X<5, то следует добавить в базу убеждений тот факт, что факториал X+1 равен (X+1)\*Y». Обратите внимание, что событие «+fact(X, Y)» является как бы шаблоном для описания множества событий, возникающих, если переменным X и Y приписать конкретные значения. Программа будет работать следующим образом. Благодаря наличию начального убеждения «fact(0,1).» и первому плану, у агента будут последовательно формироваться следующие убеждения:

fact(1,1).

fact(2,2).

fact(3,6).

fact(4,24).

fact(5,120).

При возникновении последнего убеждения первый план уже не активируется, зато активируется второй план, который выведет результат вычисления факториала.

**15 слайд.**

На этом все. Спасибо за внимание. есть ли какие-нибудь вопросы?