

Объектно-ориентированная модель

- Понятие **объекта**
- Организация программы в объектно-ориентированной модели, **сообщения**
- Понятие **класса**
- Принципы ОО подхода: **инкапсуляция, наследование, полиморфизм**
- Сравнение процедурного и объектно-ориентированного подходов

Эпиграф

Самолет - это совокупность элементов, каждый из которых по своей природе стремится упасть на землю, но за счет совместных непрерывных усилий преодолевающих эту тенденцию. Он летит только благодаря согласованным усилиям своих компонентов

Понятие объекта

- **Все** является **объектом**. Любая предметная область рассматривается как **набор объектов**
- **Объект** - это конкретный опознаваемый предмет, единица или **сущность** (реальная или абстрактная), имеющая четко **определенное функциональное назначение** в данной предметной области

Примеры объектов

- **Примеры осязаемых** или видимых объектов: бильярдный шар, трехмерный графический примитив – куб, сфера и т.п.
- **Примеры неосязаемых** объектов, событий или процессов: траектория движения шара, химический процесс, линия пересечения графических примитивов

Задание

Выделить объекты в игре «шахматы»

Понятие объекта

- **Объект** — это абстракция предметной области, имеющая **интерфейс** в виде именованных **операций** и собственные **данные**, с ограничением доступа к ним
- Объект обладает **состоянием** и **поведением**

Состояние объекта

- **Состояние** объекта характеризуется перечнем всех **свойств** данного объекта и текущими **значениями** каждого из этих свойств
- **Свойство** — это характеристики, черты, качества или способности, делающие данный объект самим собой
- **Пример** состояния некоторой окружности

Свойство	Значение
центр	(0, 2)
радиус	15
цвет	«зеленый»

Задание

Определить состояние объекта «ферзь»

Поведение объекта

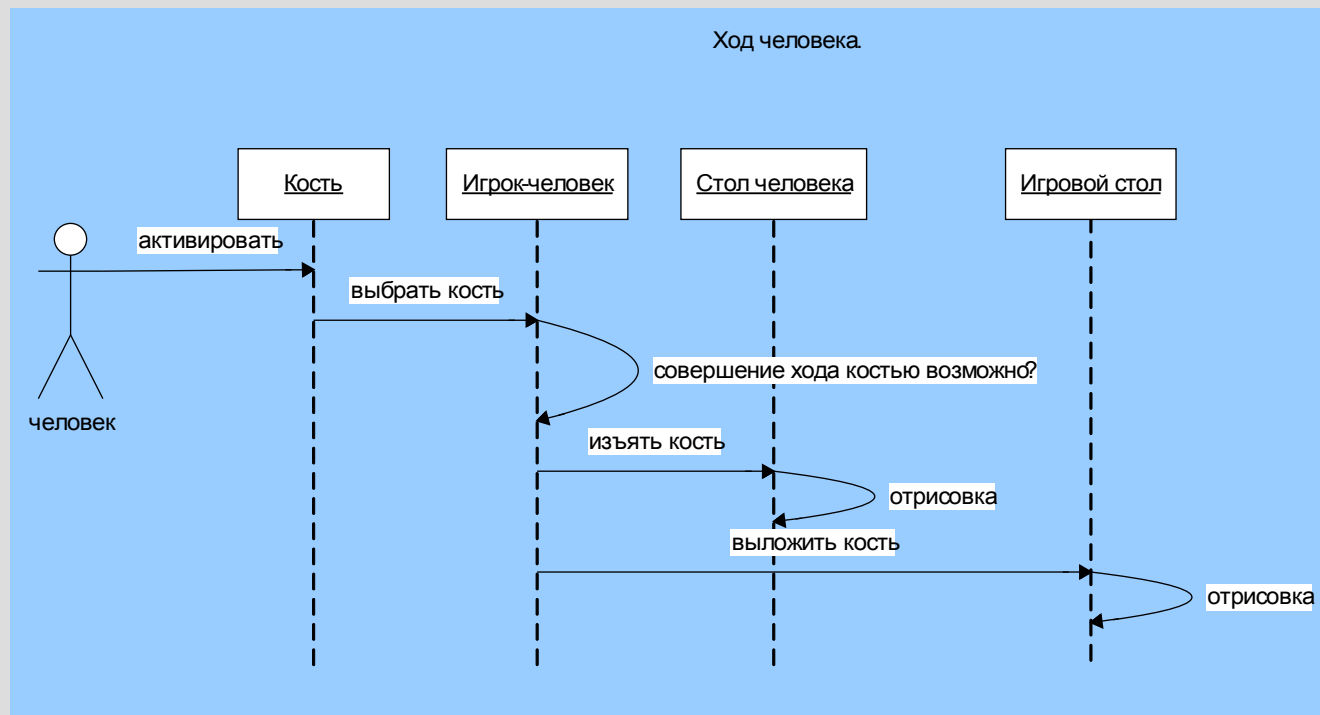
- **Состояние** объекта может **измениться** только в результате **вызова** методов
- **Поведение** – это то, как объект действует или реагирует
- **Поведение** – это набор **операций** (**методов**) объекта
- **Операция** – это услуга, которую объект может предоставить своим клиентам (другим объектам)

Задание

Определить поведение объекта «ферзь»

Организация программы с точки зрения объектно-ориентированной модели

- **Вычисление** осуществляется путем **взаимодействия** между **объектами**, при котором один объект (**клиент**) требует, чтобы другой объект (**получатель** или сервер) выполнил некоторое действие



Понятие сообщения

- Объекты **взаимодействуют** друг с другом, посылая и получая **сообщения**
- **Сообщение** - это **запрос** на выполнение действия, дополненный набором аргументов, которые могут понадобиться при выполнении действия
- В качестве **реакции** на сообщение получатель запустит некоторый **метод**, чтобы удовлетворить заданный запрос

Отличие сообщений от методов

- У **сообщения** имеется конкретный **получатель**
- **Интерпретация** одного и того же сообщения **зависит** от получателя и является **различной** для различных получателей
- Получатель может **по разному** интерпретировать (вызывать различные методы) одно и то же сообщение в зависимости от передаваемых аргументов

Задание

Выделить сообщения, которые должен интерпретировать объект «ферзь»

Пример: деятельность организации с точки зрения объектно-ориентированной модели

- С целью достижения некоторого результата **руководитель** организации **издает** письменное или устное **распоряжение**, которое поступает к начальникам подразделений
- Те в свою очередь «**транслируют**» **распоряжения исполнителям**. После выполнения или невыполнения требуемых действий наверх **сообщается** о достигнутых результатах
- На основе этих результатов **руководитель** организации **снова издает распоряжение** и т.д.

Пример: деятельность организации с точки зрения объектно-ориентированной модели

Выводы:

- Большая часть работы **руководителей** – это **посылка** **разного рода сообщений**
- **Исполнители** кроме пересылки сообщений **должны** **выполнять** **какую-то работу**

Понятие класса

- **Класс** – это некоторое **множество объектов**, имеющих **общую** структуру и общее поведение. Любой **объект** является **экземпляром** класса
- Все экземпляры одного класса будут вести себя **одинаковым** образом в ответ на одинаковые запросы
- **Класс** — множество объектов с общей структурой и поведением

Задание

Выделить классы в игре «шахматы»

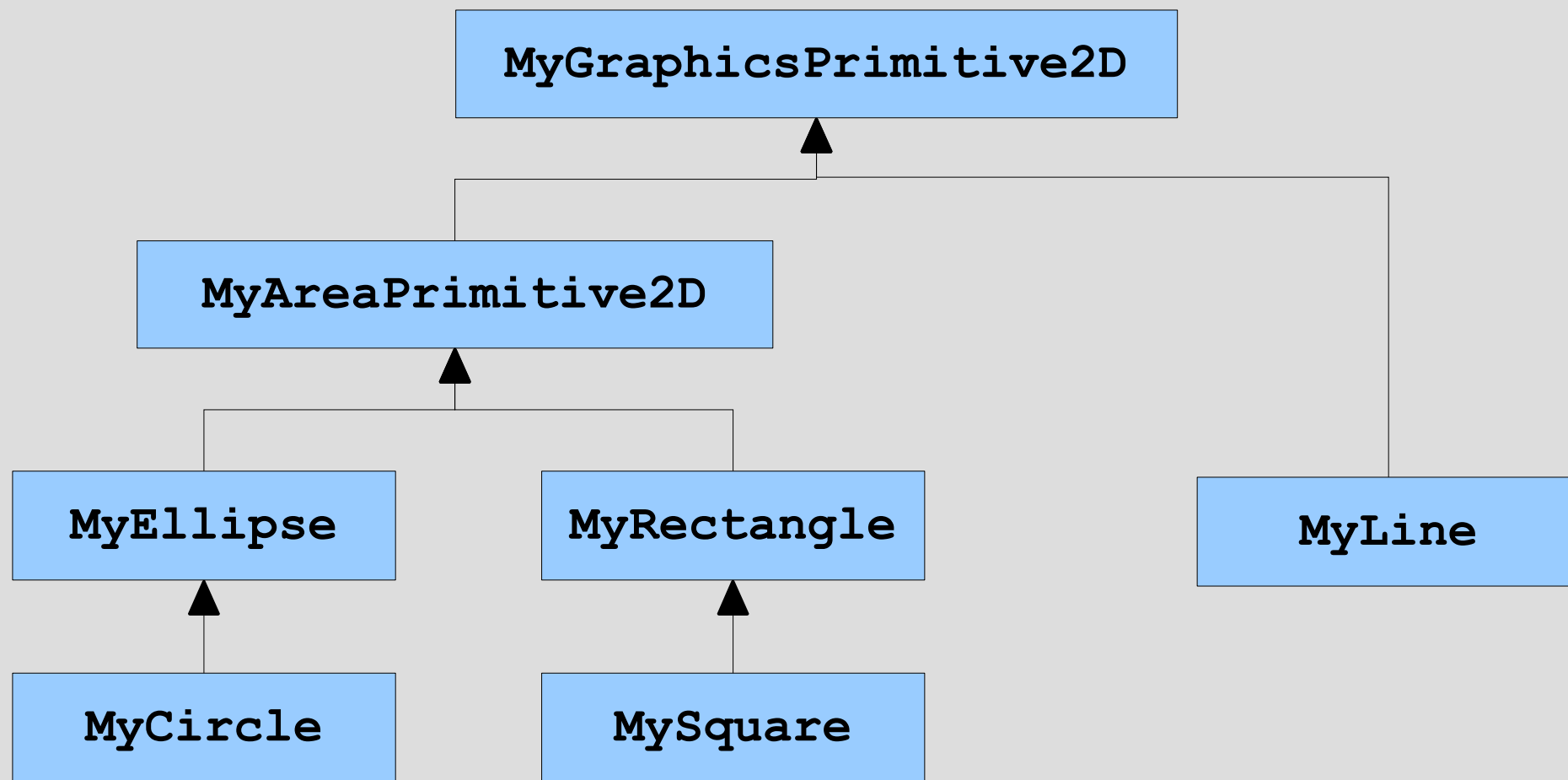
Принципы объектно-ориентированного подхода

- Класс должен проектироваться и разрабатываться с использованием принципов **инкапсуляции**, **наследования** и **полиморфизма**
- Правильное применение указанных принципов повышает **абстрагирование** и улучшает **классификацию** предметной области (программы), что в конечном счете позволят **бороться со сложностью** предметной области

Понятие абстрагирования и классификации

- **Абстрагирование** (abstraction) — это представление предметной области (программы) в виде **меньшего** количества более крупных понятий (блоков) и **минимизация** связей между ними
- **Классификация** — это выделение **существенных, устойчивых отношений** (связей) между понятиями (блоками)

Примеры абстрагирования и классификации



Задание

Выделить классы в игре «шашки»

Понятие инкапсуляции

- **Инкапсуляция** - это **объединение** данных с функциями их обработки в сочетании с **сокрытием** ненужной для использования этих данных информацией
- **Инкапсуляция** - это **сокрытие реализации** класса и **отделение** его внутреннего представления от внешнего (**интерфейса**)

Понятие интерфейса и реализации

- **Интерфейс** (interface) - это **внешний вид** класса, выделяющий его существенные черты и не показывающий внутреннего устройства и секретов поведения
- **Реализация** (implementation) - **внутреннее представление** класса, включая секреты его поведения

Задание

Определить интерфейс класса «фигура»

Преимущества использования инкапсуляции

- **Повышает степень абстракции** программы — для написания программы не требуется знания данных класса и реализация его функций
- **Позволяет изменить реализацию** класса без модификации остальной части программы, если интерфейс остался прежним

Понятие наследования

- **Наследование** – это такое отношение между классами, когда один класс **повторяет структуру и поведение** другого класса . Другими словами, структура и поведение передается от предка к потомку
- **Наследование** реализует отношение "**is-a**" между двумя классами, т.е. **дочерний** класс должен быть **частным** или специализированным случаем **родительского** класса

Задание

Применительно к игре «шахматы» определить родительский и дочерний классы

Преимущества использования наследования

- При наследовании общие свойства и поведение не описываются, что **сокращает** объем программы
- Выделение общих черт различных классов в один класс-предок является мощным **механизмом абстракции и классификации**

Понятие полиморфизма

- **Полиморфизм** — это использование **одного имени** для **различных сущностей**. При этом разнородные сущности, выступая под одним именем, воспринимаются как **однотипные**
- **Полиморфизм** — это возможностью обработки данных **переменного** типа
- **Полиморфизм** — это возможность оперировать объектами, **не** обладая **точным** знанием их типов

Задание

Привести примеры использования полиморфизма
при реализации игры «шахматы»

Преимущества использования полиморфизма

- Позволяет записывать алгоритмы лишь однажды и затем повторно их использовать для различных типов данных, которые, возможно, еще не существуют (обобщенные алгоритмы или обобщенное программирование)
- Сужает концептуальное пространство, т.е. уменьшает количество информации, которое необходимо помнить программисту

Различия между процедурным и объектно-ориентированным подходами

- В **процедурном подходе** с помощью пошагового уточнения исходная задача **разбивается на** все более мелкие **подзадачи**, пока они не станут настолько простыми, что их можно будет реализовать непосредственно
- В **объектно-ориентированном подходе** сначала **выделяются классы**, а лишь затем определяются их методы. При этом каждый метод связан с классом и класс отвечает за их выполнение

Различия между процедурным и объектно-ориентированным подходами

- В **процедурном подходе** программа представляет собой **однородное множество процедур**
- В **объектно-ориентированном подходе** классы предоставляют **удобный механизм кластеризации методов**. Кроме того скрывают детали представления данных от любого кода, кроме своих методов. Это означает, что если ошибка программирования искажает данные, то ее легче найти

Различия между процедурным и объектно-ориентированным подходами

- В процедурном подходе невозможно получить несколько копий одного модуля. Модуль - это набор связанных данных и процедур, собранных в отдельном файле. Модуль может иметь интерфейсную часть и реализацию
- В объектно-ориентированном подходе на основе класса можно создать несколько объектов с одинаковым поведением

Образное сравнение процедурного и объектно-ориентированного подходов

- Программа с точки зрения процедурного подхода – это конвейер, перемалывающий структуры данных
- Программа с точки зрения объектно-ориентированного подхода – это совокупность взаимодействующих объектов, которые просят друг друга об одолжении

Образное сравнение процедурного и объектно-ориентированного подходов

- В объектно-ориентированном подходе в центре внимания не "порядок выполнения программы", а функциональные возможности объектов. "Порядок", в котором эти возможности будут реализованы в процессе выполнения конкретной системы, является вторичным свойством

Образное сравнение процедурного и объектно-ориентированного подходов

- В объектно-ориентированном подходе роль программиста сходна с ролью пиротехника. Он складывает дрова, следя за тем, чтобы все компоненты были готовы для компоновки и необходимые связи присутствовали. Далее он зажигает спичку и следит за огнем
- Если структура правильно подготовлена, то нет необходимости стараться предсказать последовательность возгораний. Достаточно знать, что каждая часть, которая должна вспыхнуть, загорится и это произойдет не раньше положенного времени

Преимущества объектно-ориентированного подхода

- Более эффективная борьба со сложностью программного обеспечения
- Более высокий процент повторного использования кода
- Повышение надежности программного обеспечения
- Обеспечение возможности модификации отдельных компонентов программного обеспечения без изменения остальных его компонентов

Преимущества объектно-ориентированного подхода

- Использование объектного подхода существенно **повышает уровень унификации** разработки и пригодность для **повторного использования** не только программ, но и проектов, что в конце концов ведет к созданию среды разработки
- Объектно-ориентированные системы часто получаются **более компактными**, чем их не объектно-ориентированные эквиваленты. А это означает не только **уменьшение объема кода** программ, но и **удешевление проекта** за счет использования предыдущих разработок, что дает выигрыш в стоимости и времени

Преимущества объектно-ориентированного подхода

- Использование объектной модели приводит к построению систем на основе **стабильных промежуточных описаний**, что упрощает процесс внесения изменений
- Это дает возможность **развиваться постепенно** и не приводит к полной ее переработке даже в случае существенных изменений исходных требований

Преимущества объектно-ориентированного подхода

- Объектная модель **уменьшает риск разработки** сложных систем, прежде всего потому, что процесс интеграции растягивается на все время разработки, а не превращается в единовременное событие
- Объектный подход состоит из **ряда** хорошо **продуманных** этапов проектирования, что также уменьшает степень риска и повышает уверенность в правильности принимаемых решений

Преимущества объектно-ориентированного подхода

- Объектная модель ориентирована на **человеческое восприятие** мира: многие люди, не имеющие понятия о том, как работает компьютер, находят вполне естественным объектно-ориентированный подход к системам