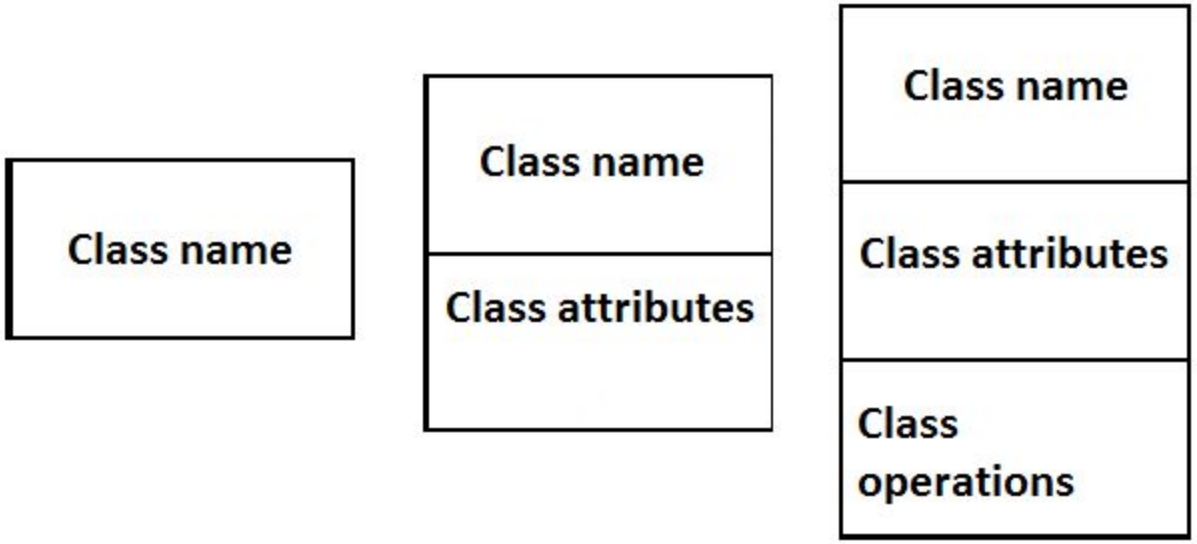
Классы и объекты

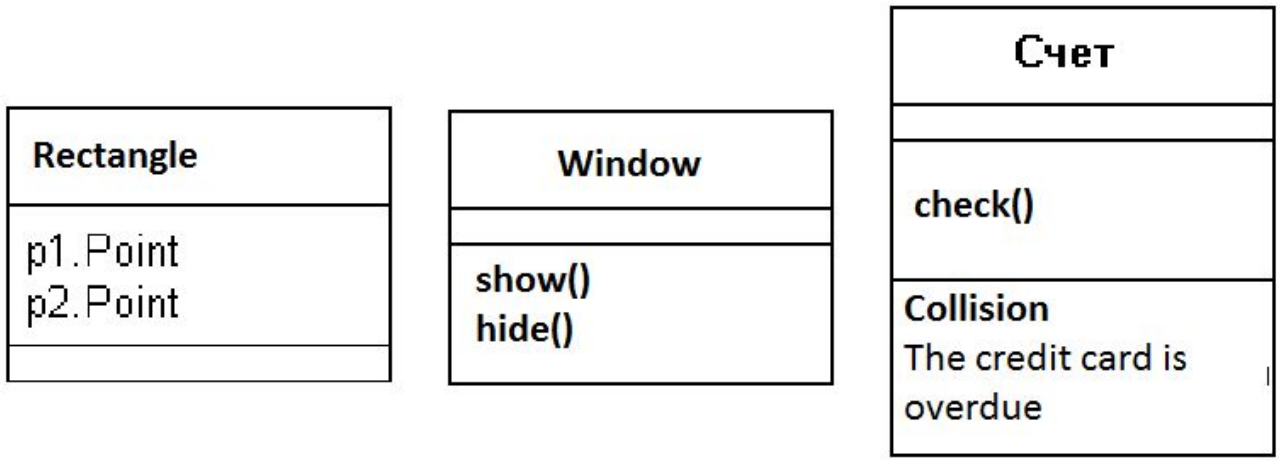
Основные термины

- Классификатор
- Класс
- Интерфейс
- Экземпляр класса
- Ассоциация
- Квалификатор
- Класс ассоциации
- Обобщение
- Украшение
- Тип данных
- Пакеты
- Отношение доступа
- Отношение импорта
- Отношение объединения
- Ограничение

Класс


Class name
Class name
Class attributes
Class name
Class attributes
Class operations

Класс


Rectangle
p1.Point
p2.Point
Window
show()
hide()
Счет
check()
Collision
The credit card is overdue

Кванторы видимости

“+” – public

“#” – protected

“–” – private

Мультипликативность

- [0..1]
- [0..*]
- [1..*]
- [1..5]
- [1..3, 5, 7]
- [1..3, 7.. 10]
- [1..3,7..*]

Атрибуты класса

- color: Color
- employee_name[1..2]: String
- visibility: Boolean
- form: Polygon

Атрибуты класса

- color: Color = (255, 0, 0)
- employee_name[1..2]: String = “Ivan Ivanovich”
- visibility: Boolean = true
- form: Polygon = rectangle
- visibility: Boolean = false

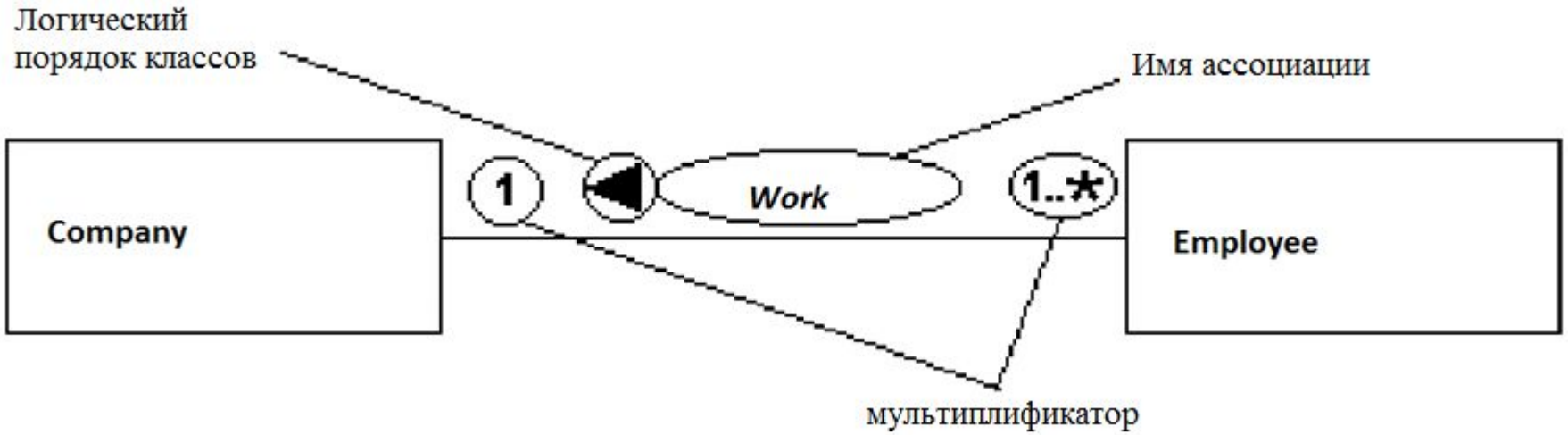
Операции класса

- +create()
- +draw(form: Polygon = rectangle; fill_color: Color = (0, 0, 255))
- ask_client’s_account(number_of_account: Integer): Currency
- show_message():{"division by zero Error"}

Операции ассоциации

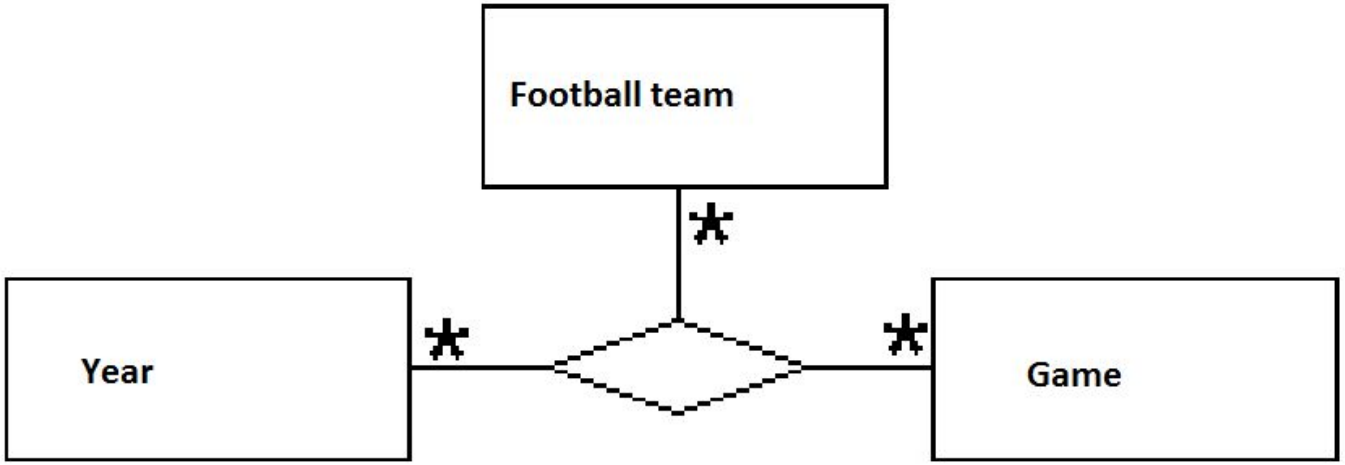
- Соответствуют наличию некоторого отношения между классами, возможности одного класса узнавать о публичных атрибутах и операциях другого класса
- Ассоциация может быть направленной и ненаправленной. Направленная ассоциация может быть однонаправленной и двунаправленной.
- Если ассоциация однонаправленная, то один класс-участник должен знать о другом, тот, в свою очередь, знать о первом классе не обязан и может использоваться без него. Например: «Законодательство» – «Гражданин». Каким другим отношением можно заменить такую ассоциацию?
- Если ассоциация двунаправленная, то каждый класс-участник должен знать о другом и ни один из них не может использоваться без другого. Например: «Человек» – «Автомобиль»
- Что означает ненаправленная ассоциация?

Ассоциация

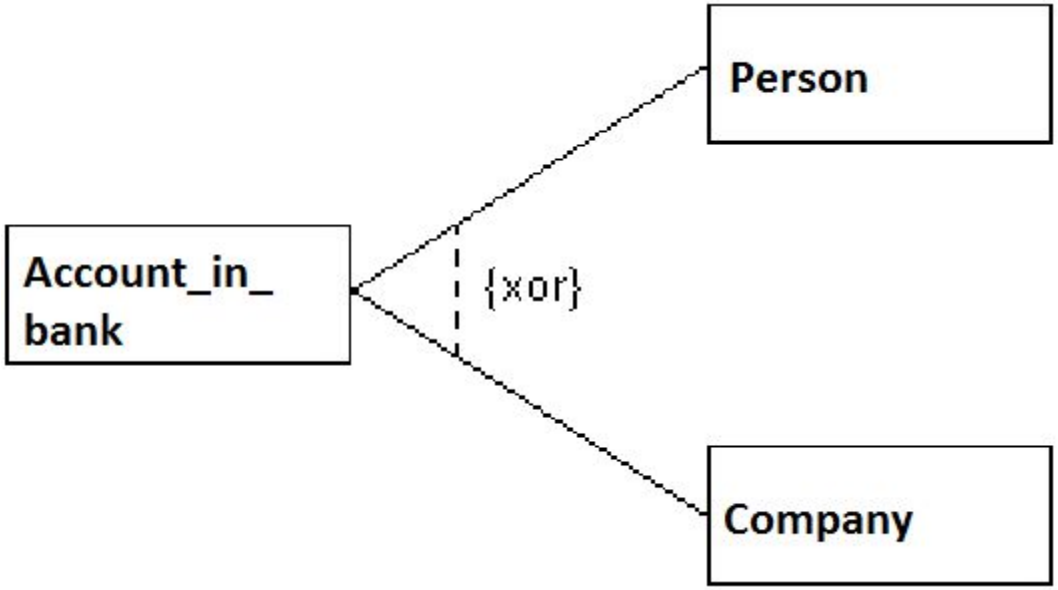


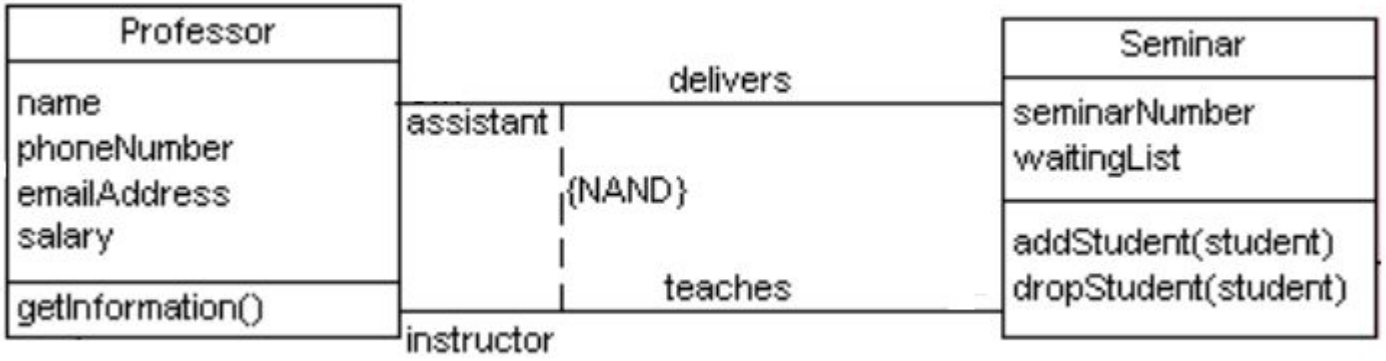
Любая ассоциация обладает двумя ролями: каждая роль представляет собой направление ассоциации.

Ассоциация

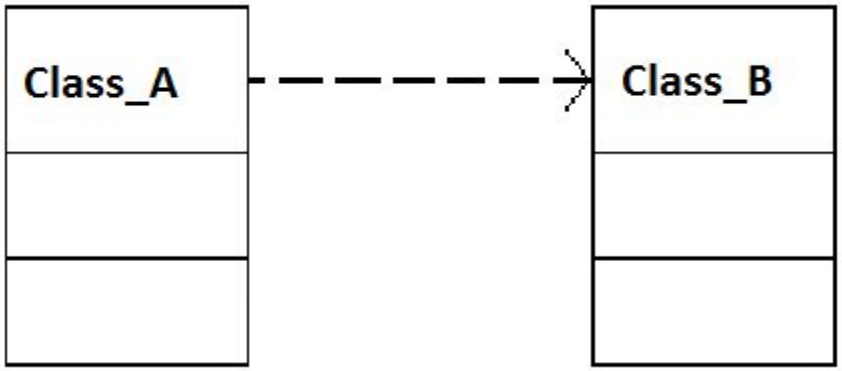


Ассоциация

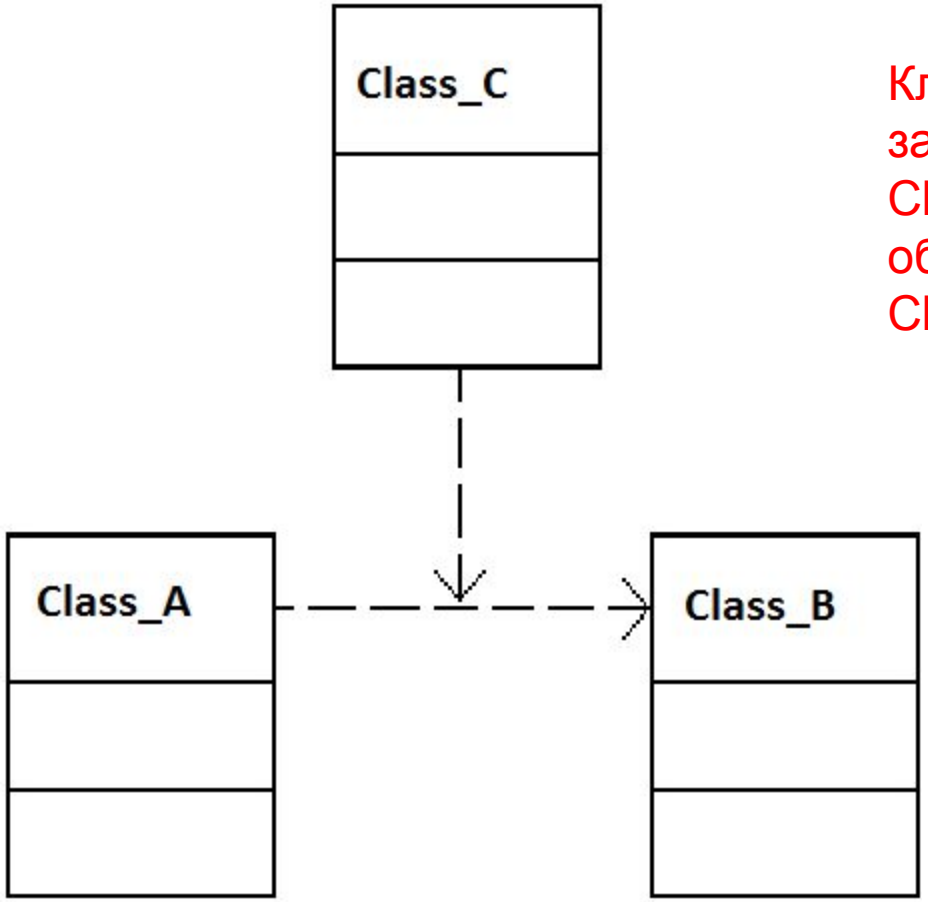

Person
Account_in_
bank
{xor}
Company


Professor
name
phoneNumber
emailAddress
salary
getInformation()
delivers
assistant
{NAND}
teaches
instructor
Seminar
seminarNumber
waitingList
addStudent(student)
dropStudent(student)

Отношение зависимости



Отношение зависимости



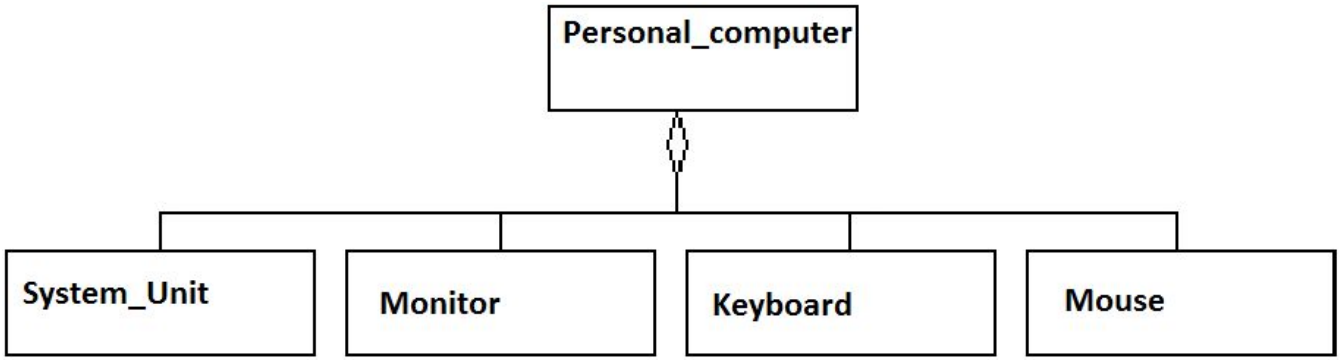
Класс Class_C зависит от класса Class_B или от обоих классов Class_A и Class_B?

Стереотипы отношения зависимости

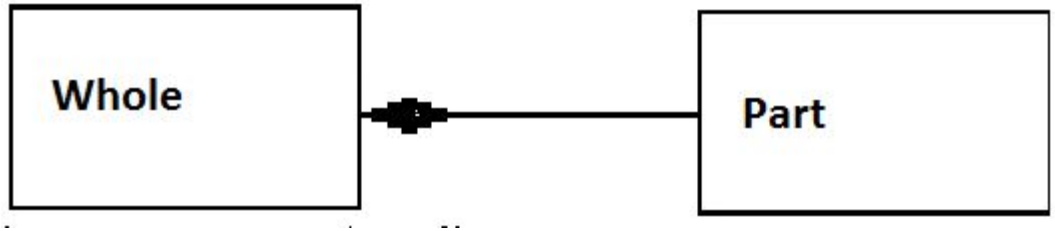
- “access” – один класс обеспечивает доступ к другому классу
- “bind” – один класс связан с другим классом
- “derive” – зависимый класс может быть восстановлен по информации независимого класса. Стереотип применим также к атрибутам и операциям. Зависимый элемент излишен, введен для удобства, наглядности и т.д.
- “import” – один класс включает другой класс
- “refine” – один класс уточняет другой класс. Связанные классы концептуально совпадают, но находятся на разных уровнях абстракции
- “call” – операция зависимого класса вызывает операцию независимого класса

- Приведите примеры на разные виды стереотипы зависимости.
- Какими другими типами отношений между классами можно заменить некоторые стереотипы зависимости?

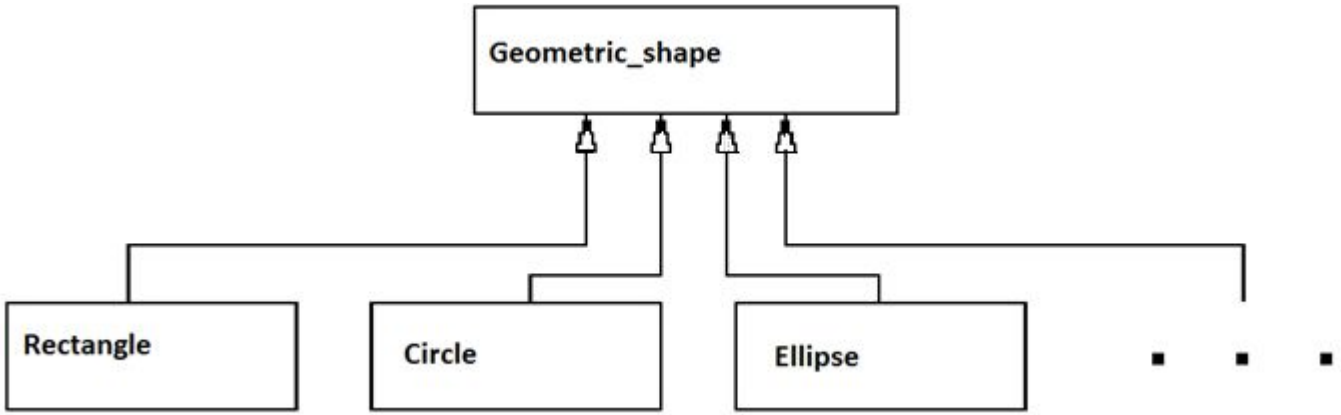
Агрегация



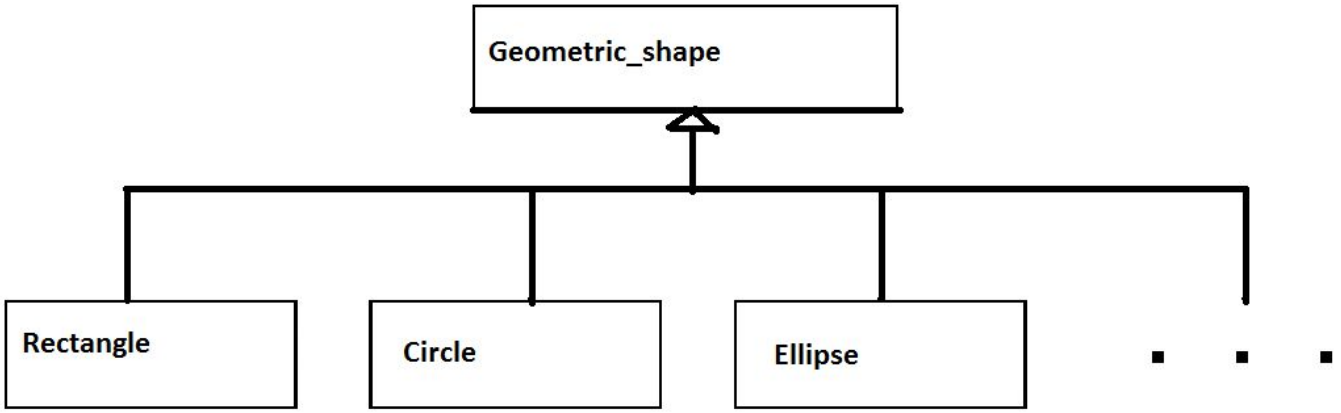
Композиция



Обобщение



Обобщение

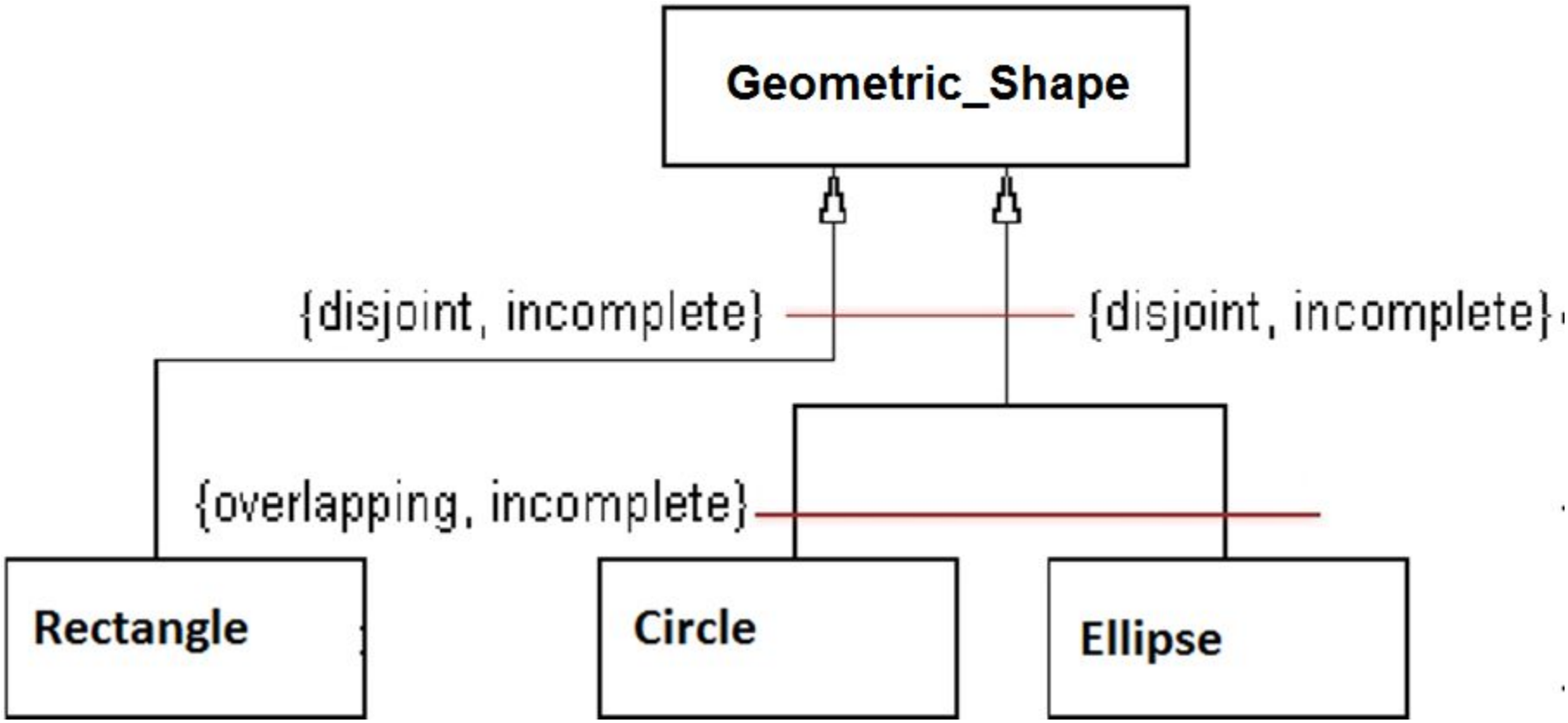


Ограничения

- {complete} – полный перечень подчиненных классов, других не существует
- {incomplete} – перечень подчиненных классов не завершен

- {disjoint} – непересекающиеся классы
- {overlapping} – перекрытие классов

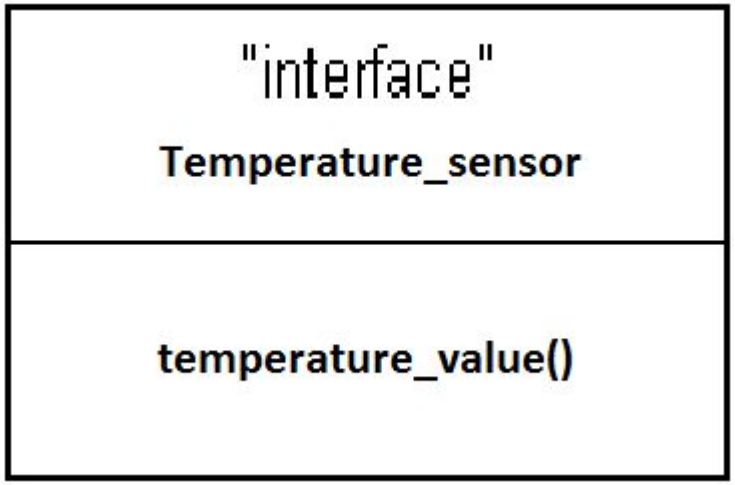
Обобщение


Geometric_Shape
{disjoint, incomplete}
{disjoint, incomplete}
{overlapping, incomplete}
Rectangle
Circle
Ellipse

Обобщение

- Смысл обобщения (уточнения, наследования): интерфейс подтипа должен включать все элементы интерфейса супертипа. Другая сторона обобщения – принцип подстановочности.
- Пример: «Исполнитель» обобщает «Субподрядчика». Субподрядчика можно подставить в любой код, где требуется «Исполнитель», и при этом все должно нормально работать. Любой экземпляр любого подтипа «Исполнителя» должен свободно работать на коде, предполагающем использование «Исполнителя». Субподрядчик может реагировать на некоторые команды отличным от другого исполнителя образом (в соответствии с каким принципом ООП?), но это отличие не должно беспокоить вызывающий объект.
- Какой принцип методологии SOLID заложен в семантике отношения обобщения?

Интерфейс



- Если класс реализует интерфейс, то отношение между ним и интерфейсом – реализация.
- Если класс использует интерфейс, то отношение между ним и интерфейсом – зависимость со стереотипом “call”.

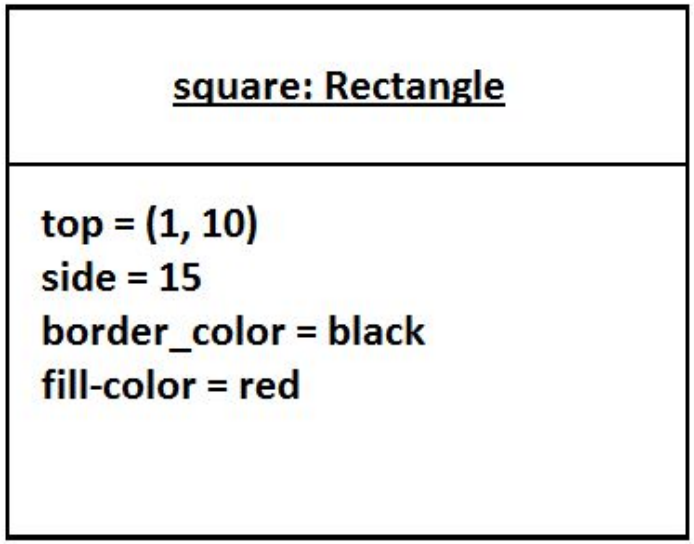
Интерфейс

- В блоке с названием класса, интерфейса может быть указано ключевое слово в кавычках (например, “interface”) или фигурных скобках (например, {abstract}).
- В кавычках пишем то, что указывает на метакласс, определенный в UML.
- В фигурных скобках пишем то, что указывает на значение свойства класса. Например, {abstract} указывает на значение свойства isAbstract метакласса Class.

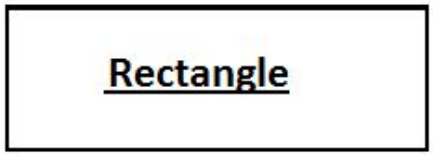
Объекты

(a)

(b)

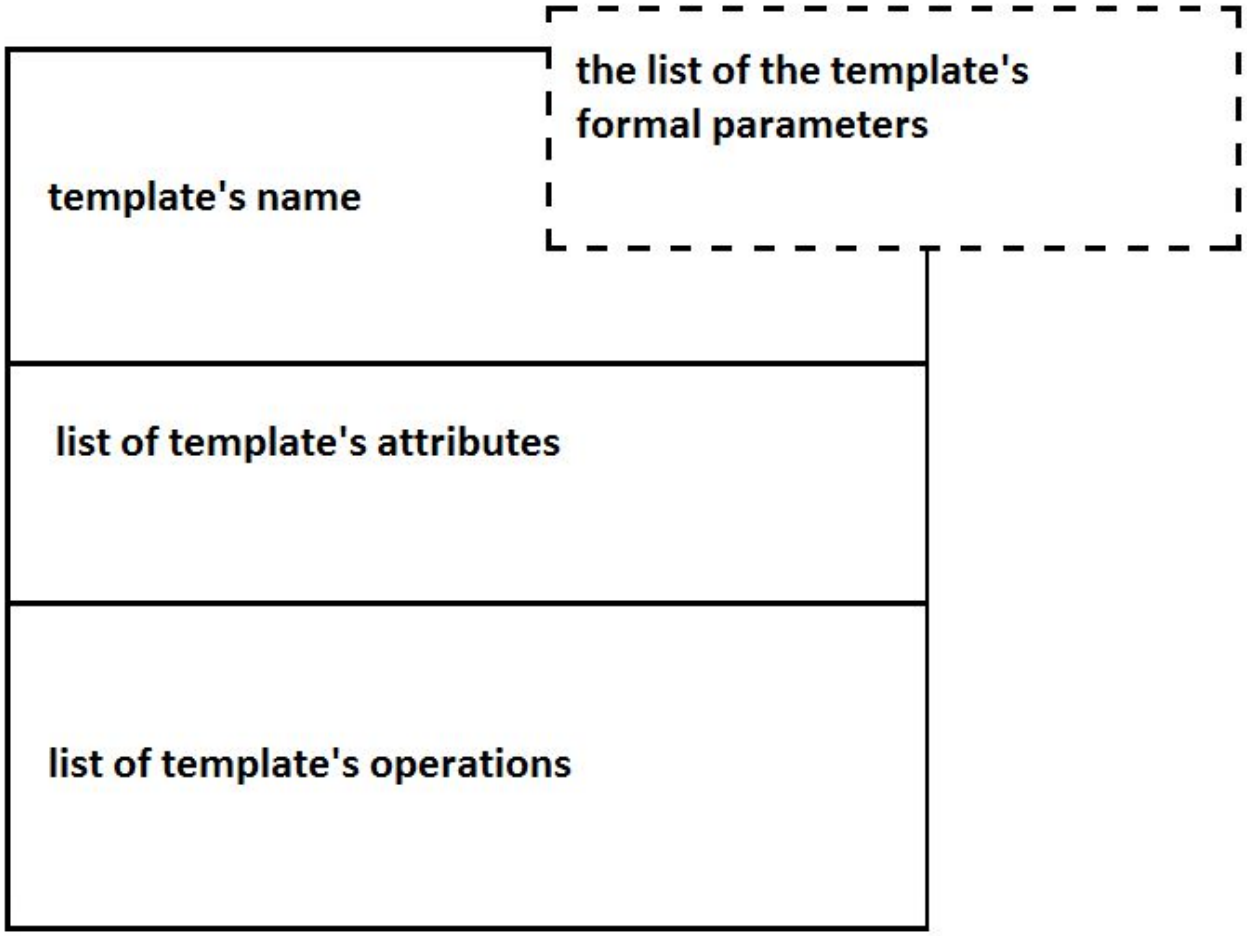


(c)

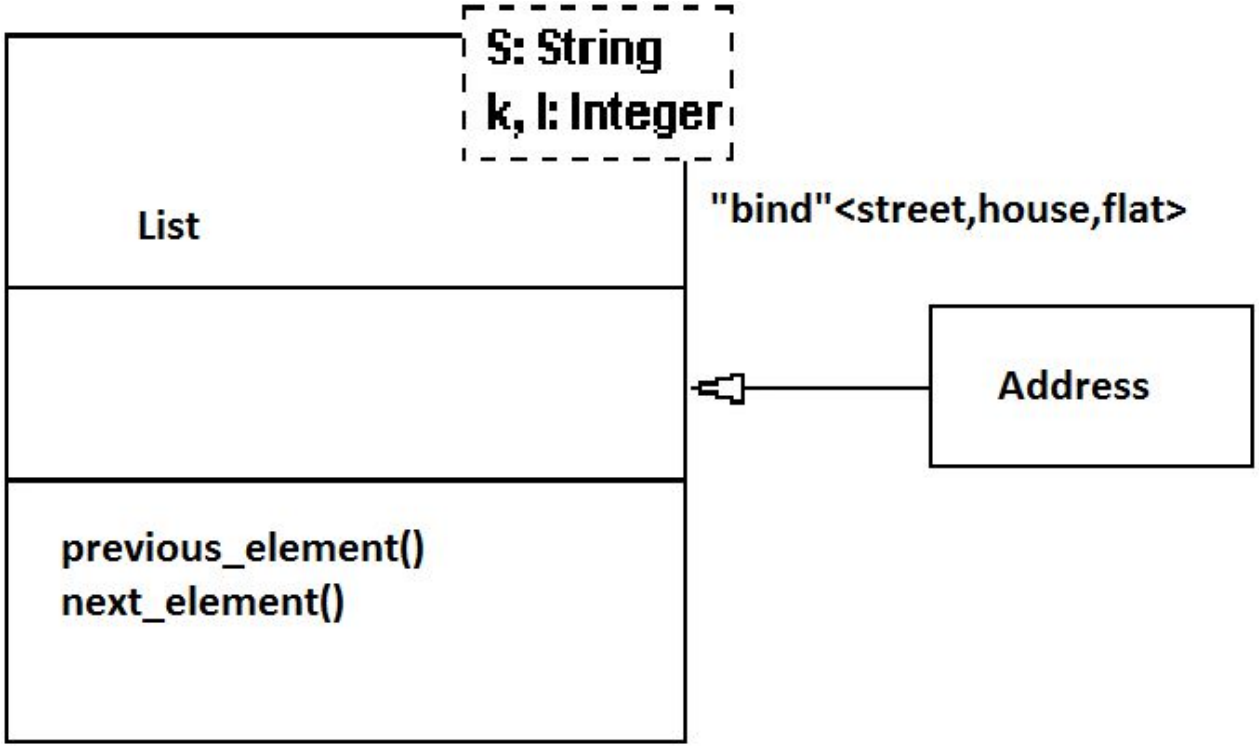


(d)

Шаблоны


the list of the template's formal parameters
template's name
list of template's attributes
list of template's operations

Шаблоны


S: String
k, l: Integer
List
"bind"<street,house,flat>
Address
previous_element()
next_element()

Рекомендации по созданию диаграммы классов

Хорошо структурированная (согласованная, well-formed) диаграмма классов:

- фокусируется только на одном аспекте статического представления системы;
- содержит только те элементы, которые существенны для понимания данного аспекта;
- предоставляет информацию в соответствии с уровнем их абстракции, только с теми украшениями, которые необходимы для понимания;
- не настолько минималистична, чтобы дезинформировать читателя о важных семантических особенностях.

Рекомендации по созданию диаграммы классов

Когда вы создаете диаграмму классов:

- давайте ей имя в соответствии с ее назначением;
- располагайте элементы так, чтобы минимизировать пересечение линий;
- организуйте элементы в пространстве так, чтобы семантически связанные сущности располагались близко друг к другу;
- используйте замечания и выделения цветом как визуальные подсказки, чтобы обратить внимание на важные особенности диаграммы;
- Старайтесь не использовать слишком много разных типов отношений. В общем случае, один вид отношений должен доминировать на диаграмме классов.

Вопросы для самостоятельного изучения

- Сравнить CASE-средства BPWin, ERWin, Rational Rose, ATS по следующим критериям: технология моделирования, тип модели информационной системы в результате, способ создания модели, представления, поддерживаемый язык, автоматическое создание БД, автоматическая генерация кода приложения

Вопросы для самостоятельного изучения

- Узнать наиболее известные CASE-средства следующих типов:
  - Только графические ОО CASE-средства
  - ОО CASE-средства с частичной кодогенерацией (от 1% до 60%)
  - Мета ОО CASE-средства
  - ОО CASE-средства полного цикла разработки
- Узнать, зачем нужны методы нормализации диаграммы классов:
  - Применение паттернов
  - Рефакторинг

Вопросы для самостоятельного изучения

- Как на диаграмме классов обозначается класс ассоциации и зачем он нужен?