

Объектно-ориентированное программирование

Марков Евгений Михайлович к.т.н., доцент,
доцент кафедры «ВТ»

Список литературы:

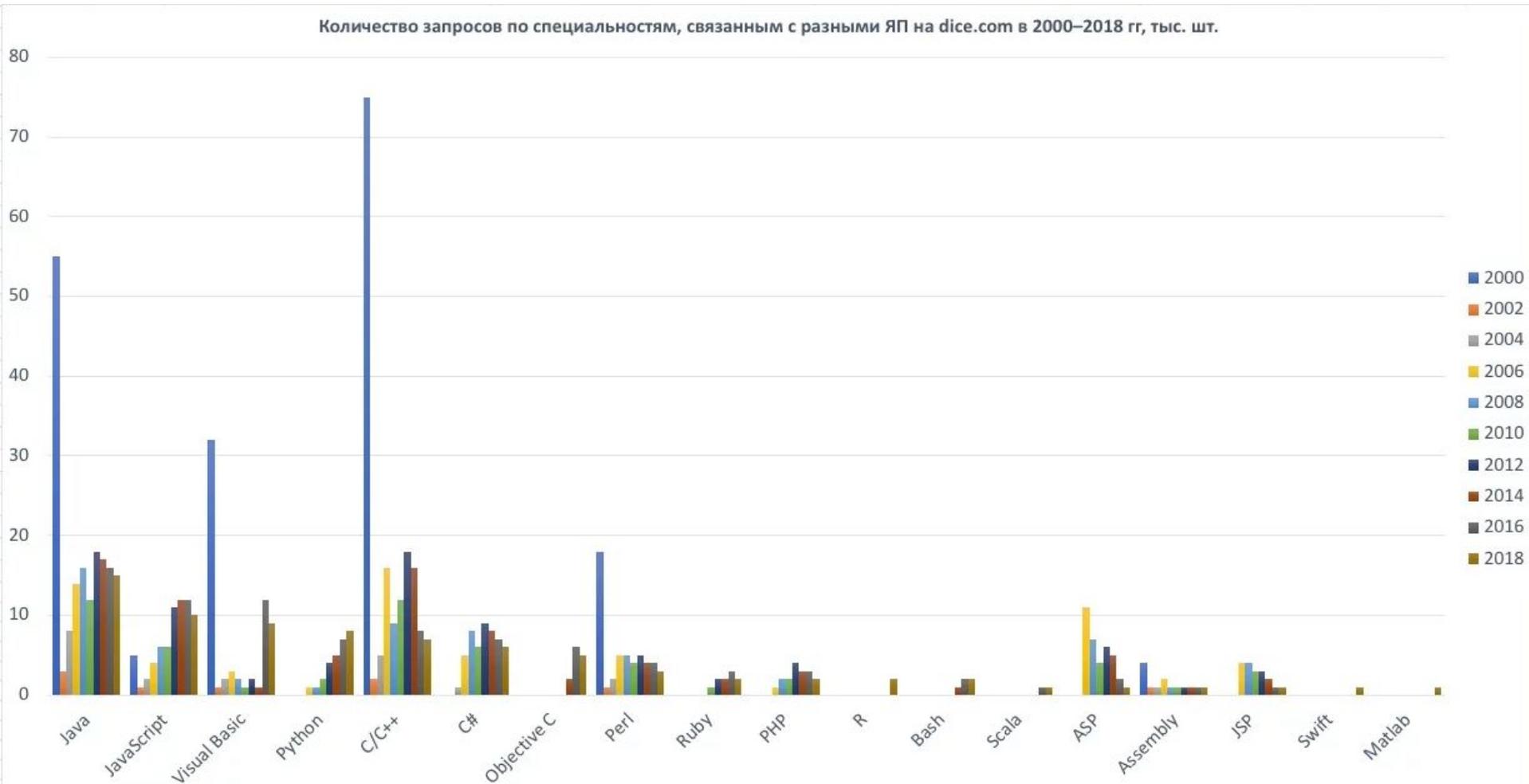
Р. ЛАФОРЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ В C++ 4-Е
ИЗДАНИЕ

Бертран Мейер Объектно-ориентированное конструирование программных
систем

Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-
ориентированного проектирования. Паттерны проектирования.

Вайсфельд М. Объектно-ориентированное мышление.

Популярность и назначение языков программирования



назначение языков программирования

PYPL Popularity of
Programming Language
2020

Rank	Change	Language	Share	Trend
1		Python	31.59 %	+3.3 %
2		Java	16.9 %	-2.7 %
3		Javascript	8.17 %	+0.0 %
4		C#	6.54 %	-0.7 %
5		C/C++	5.88 %	+0.1 %
6		PHP	5.78 %	-0.7 %
7		R	4.18 %	+0.3 %
8		Objective-C	2.6 %	-0.0 %
9		Swift	2.35 %	-0.0 %
10		TypeScript	1.94 %	+0.2 %
11		Matlab	1.63 %	-0.2 %
12		Kotlin	1.57 %	+0.1 %
13		Go	1.39 %	+0.2 %
14		Ruby	1.22 %	-0.2 %
15		VBA	1.19 %	-0.1 %
16		Scala	0.97 %	-0.1 %
17		Rust	0.91 %	+0.3 %
18		Visual Basic	0.82 %	-0.2 %
19		Dart	0.57 %	+0.2 %
20		Ada	0.54 %	³ +0.2 %

Основные модели программирования

- Императивное программирование
 - Процедурная
 - Структурная
 - Аспектно-ориентированная
 - Объектно-ориентированная
 - Агентно-ориентированная
 - Компонентно-ориентированная
 - Прототипно-ориентированная
- Декларативное программирование
 - Функциональное программирование
 - Логическое программирование

Императивное программирование

Императивное программирование — это парадигма программирования (стиль написания исходного кода компьютерной программы), для которой характерно следующее:

- в исходном коде программы записываются инструкции (команды);
- инструкции должны выполняться последовательно;
- при выполнении инструкции данные, полученные при выполнении предыдущих инструкций, могут читаться из памяти;
- данные, полученные при выполнении инструкции, могут записываться в память.

Декларативное программирование

Декларативное программирование — это парадигма программирования, в которой задаётся спецификация решения задачи, то есть описывается, что представляет собой проблема и ожидаемый результат.

Противоположностью декларативного является императивное программирование, описывающее на том или ином уровне детализации, как решить задачу и представить результат.

В общем и целом, декларативное программирование идёт от человека к машине, тогда как императивное — от машины к человеку. Как следствие, декларативные программы не используют понятия состояния, то есть не содержат переменных и операторов присваивания.

Процедурное программирование

Процедурное программирование — программирование на императивном языке, при котором последовательно выполняемые операторы можно собрать в подпрограммы, то есть более крупные целостные единицы кода, с помощью механизмов самого языка.

Процедурное программирование является отражением архитектуры традиционных ЭВМ, которая была предложена Фон Нейманом в 1940-х годах. Теоретической моделью процедурного программирования служит абстрактная вычислительная система под названием машина Тьюринга.

Структурное программирование

Структурное программирование — методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков. Предложена в 1970-х годах Э. Дейкстрой и др.

В соответствии с данной методологией любая программа строится без использования оператора `goto` из трёх базовых управляющих структур: последовательность, ветвление, цикл; кроме того, используются подпрограммы. При этом разработка программы ведётся пошагово, методом «сверху вниз».⁸

Функциональное программирование

Функциональное программирование — раздел дискретной математики и парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних (в отличие от функций как подпрограмм в процедурном программировании).

Противопоставляется парадигме императивного программирования, которая описывает процесс вычислений как последовательное изменение состояний (в значении, подобном таковому в теории автоматов). При необходимости, в функциональном программировании вся совокупность последовательных состояний вычислительного процесса представляется явным образом, например, как список

Логическое программирование

Логическое программирование — парадигма программирования, основанная на автоматическом доказательстве теорем, а также раздел дискретной математики, изучающий принципы логического вывода информации на основе заданных фактов и правил вывода.

Логическое программирование основано на теории и аппарате математической логики с использованием математических принципов резолюций.

Объектно-ориентированное программирование

Объектно-ориентированное программирование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

Идеологически ООП — подход к программированию как к моделированию информационных объектов, решающий на новом уровне основную задачу структурного программирования: структурирование информации с точки зрения управляемости, что существенно улучшает управляемость самим процессом моделирования, что, в свою очередь, особенно важно при реализации крупных

Компонентно-ориентированное программирование

Компонентно-ориентированное программирование — это своеобразная «надстройка» над ООП, набор правил и ограничений, направленных на построение крупных развивающихся программных систем с большим временем жизни.

Программная система в этой методологии представляет собой набор компонентов с хорошо определёнными интерфейсами.

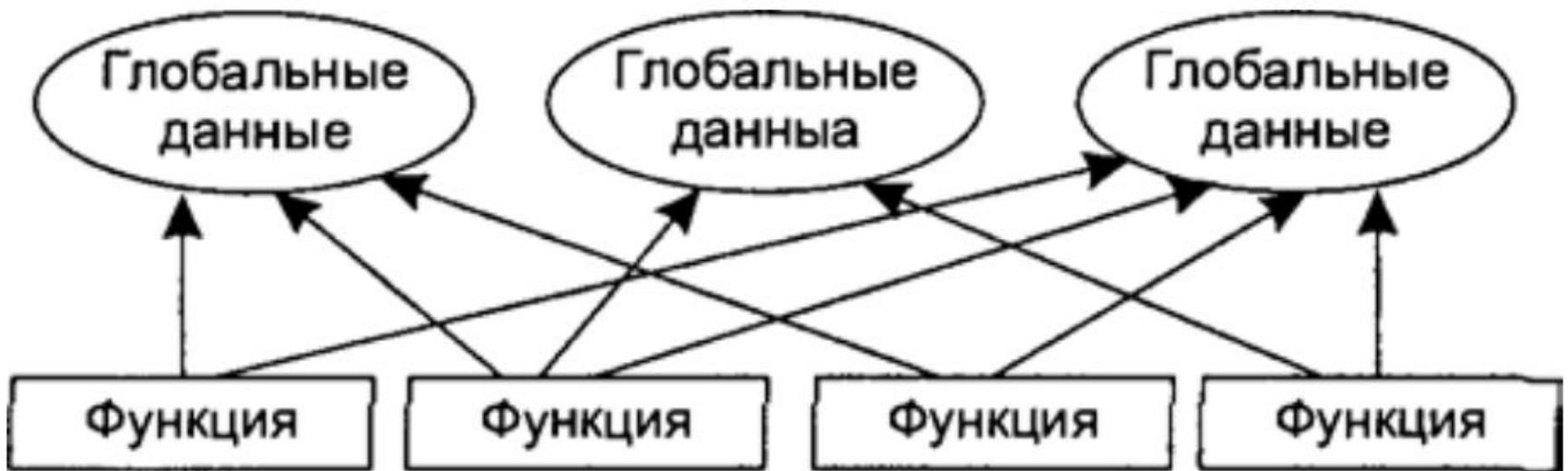
Изменения в существующую систему вносятся путём создания новых компонентов в дополнение или в качестве замены ранее существующих. При создании новых компонентов на основе ранее созданных запрещено использование наследования реализации — новый компонент может наследовать лишь интерфейсы базового. Таким образом, компонентное программирование обходит проблему хрупкости базового класса.

Неконтролируемый доступ к данным



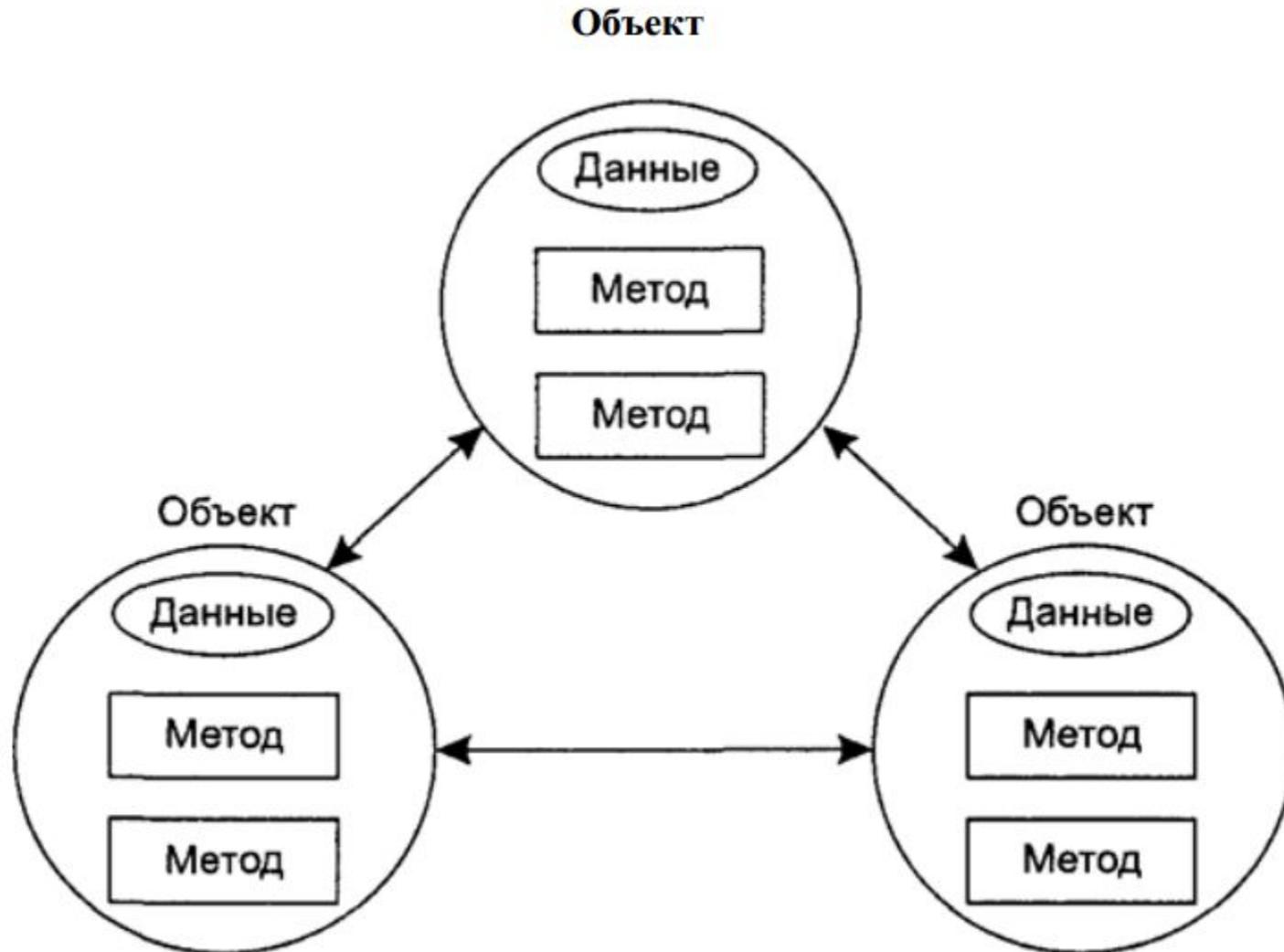
Глобальные и локальные переменные

Неконтролируемый доступ к данным



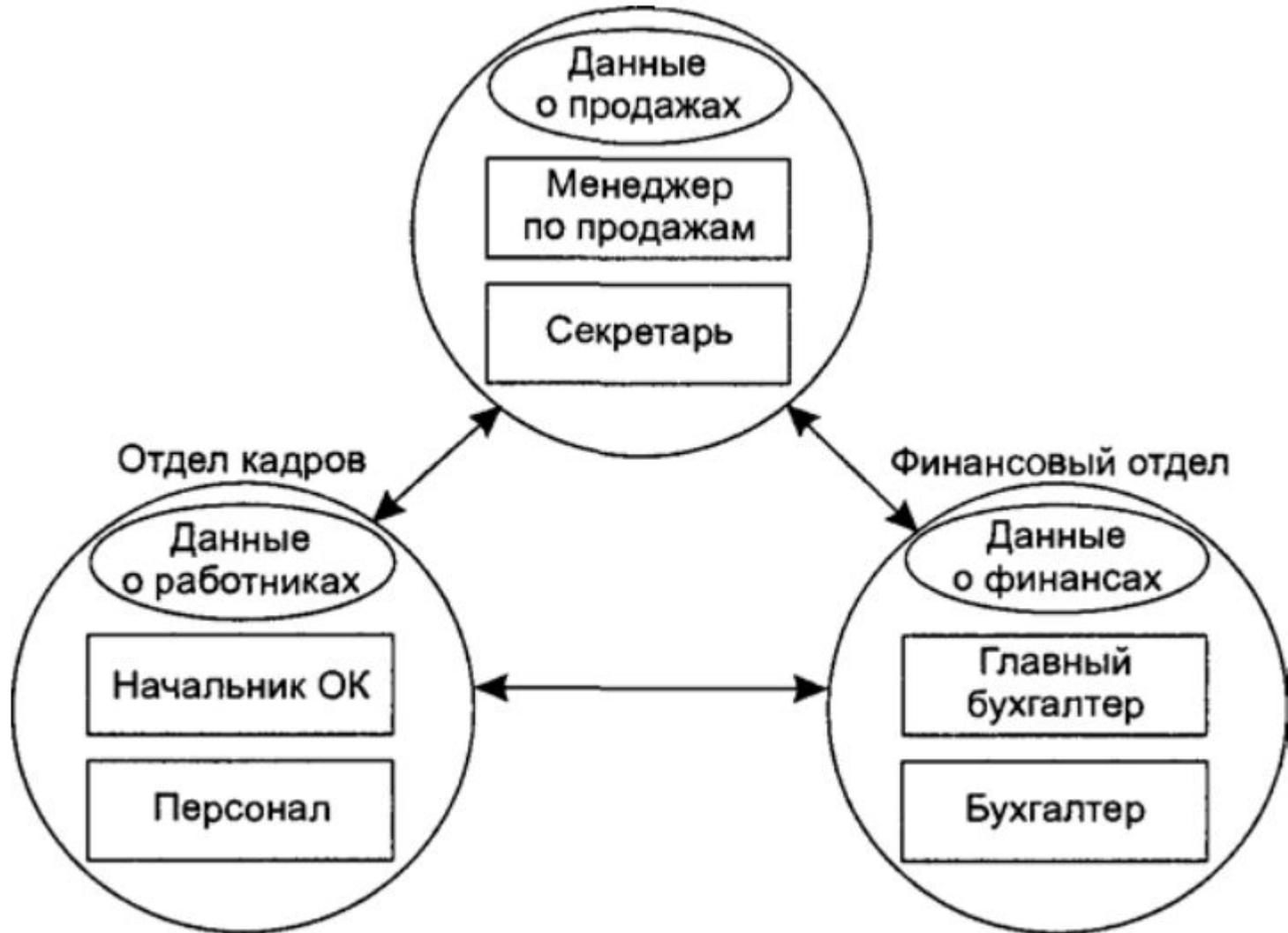
Процедурный подход

Объектно-ориентированный подход



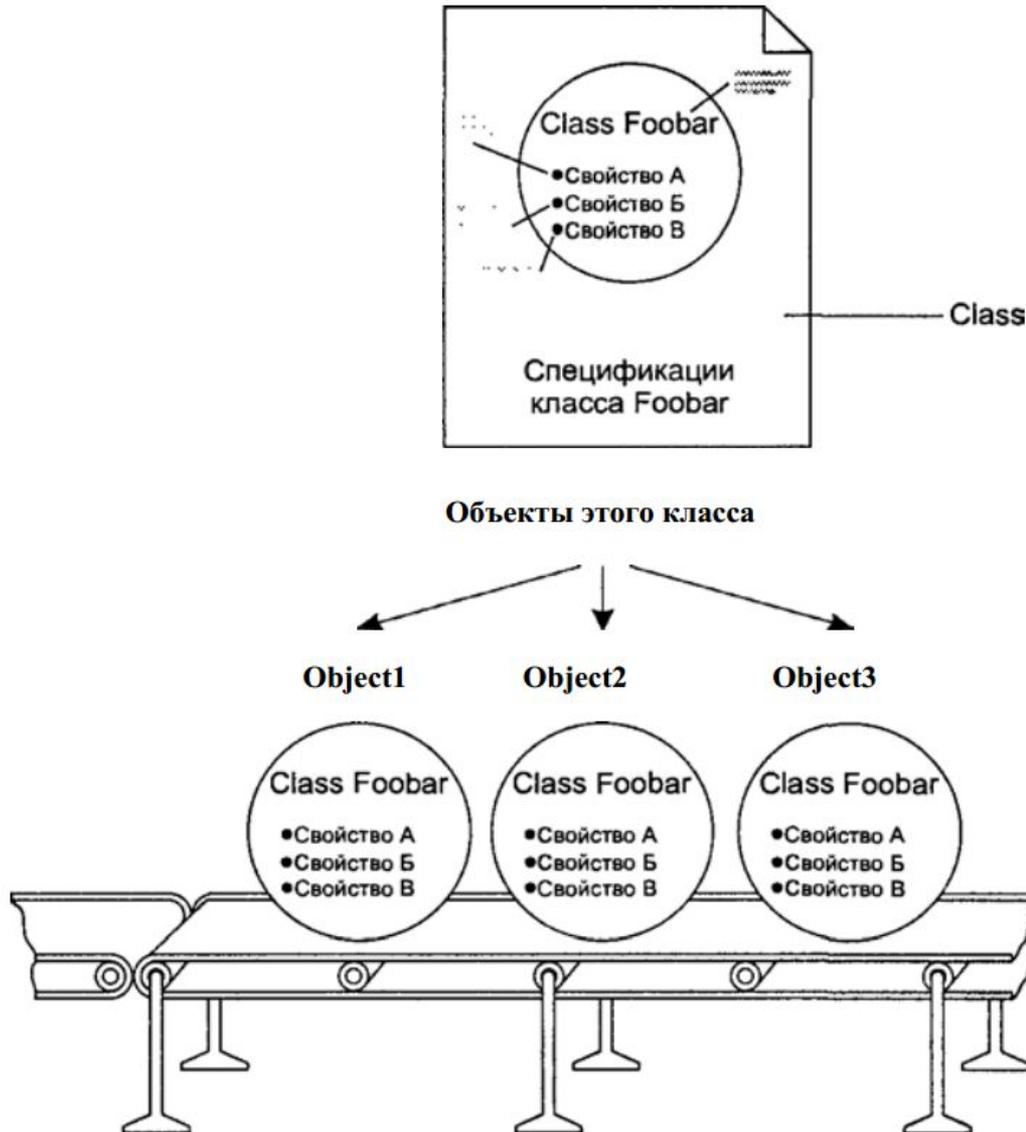
Объектно-ориентированный подход

Объектно-ориентированный подход

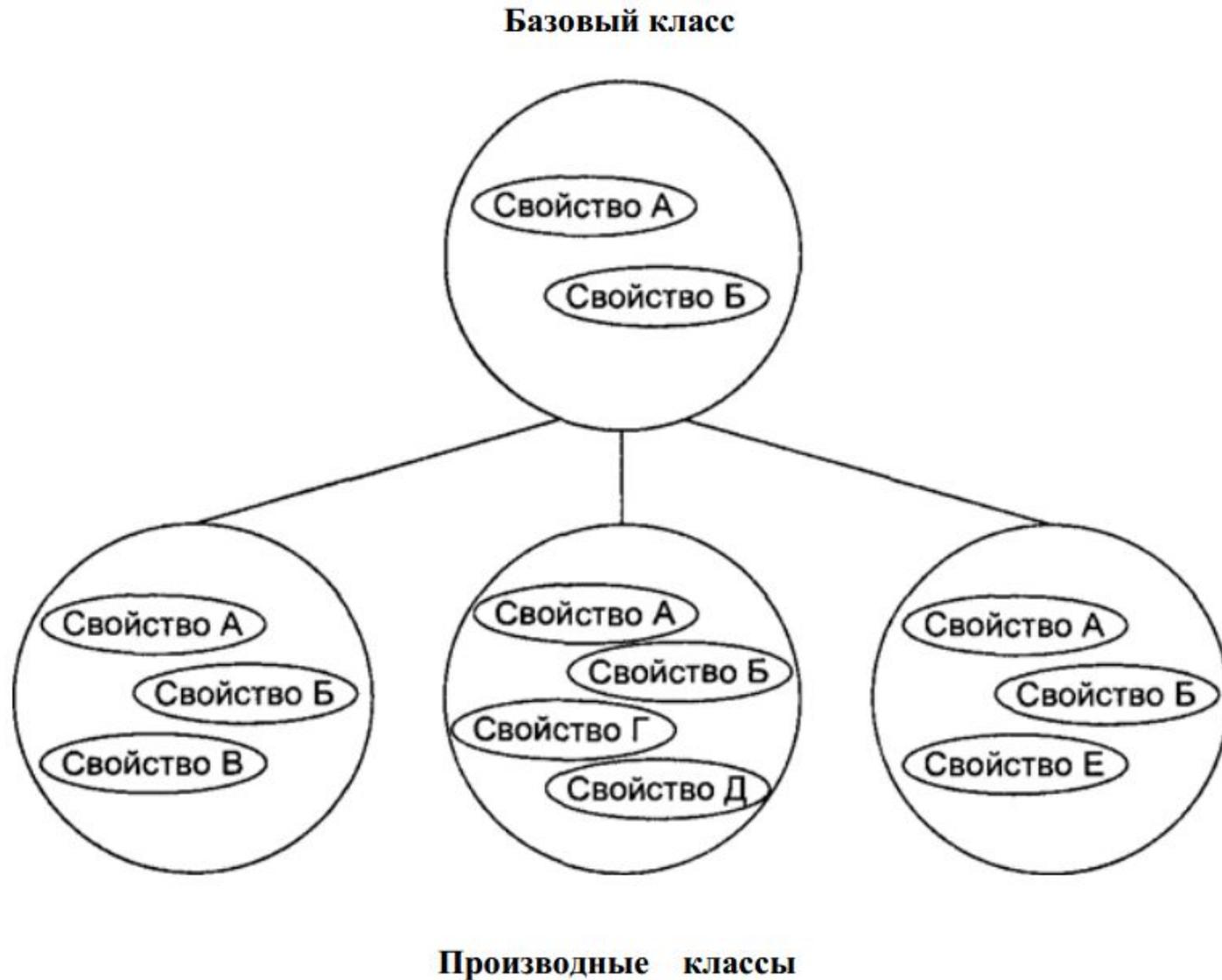


Корпоративный подход

Классы и объекты



Наследование и повторное использование кода



Полиморфизм и перегрузка



Пользовательские типы и операции

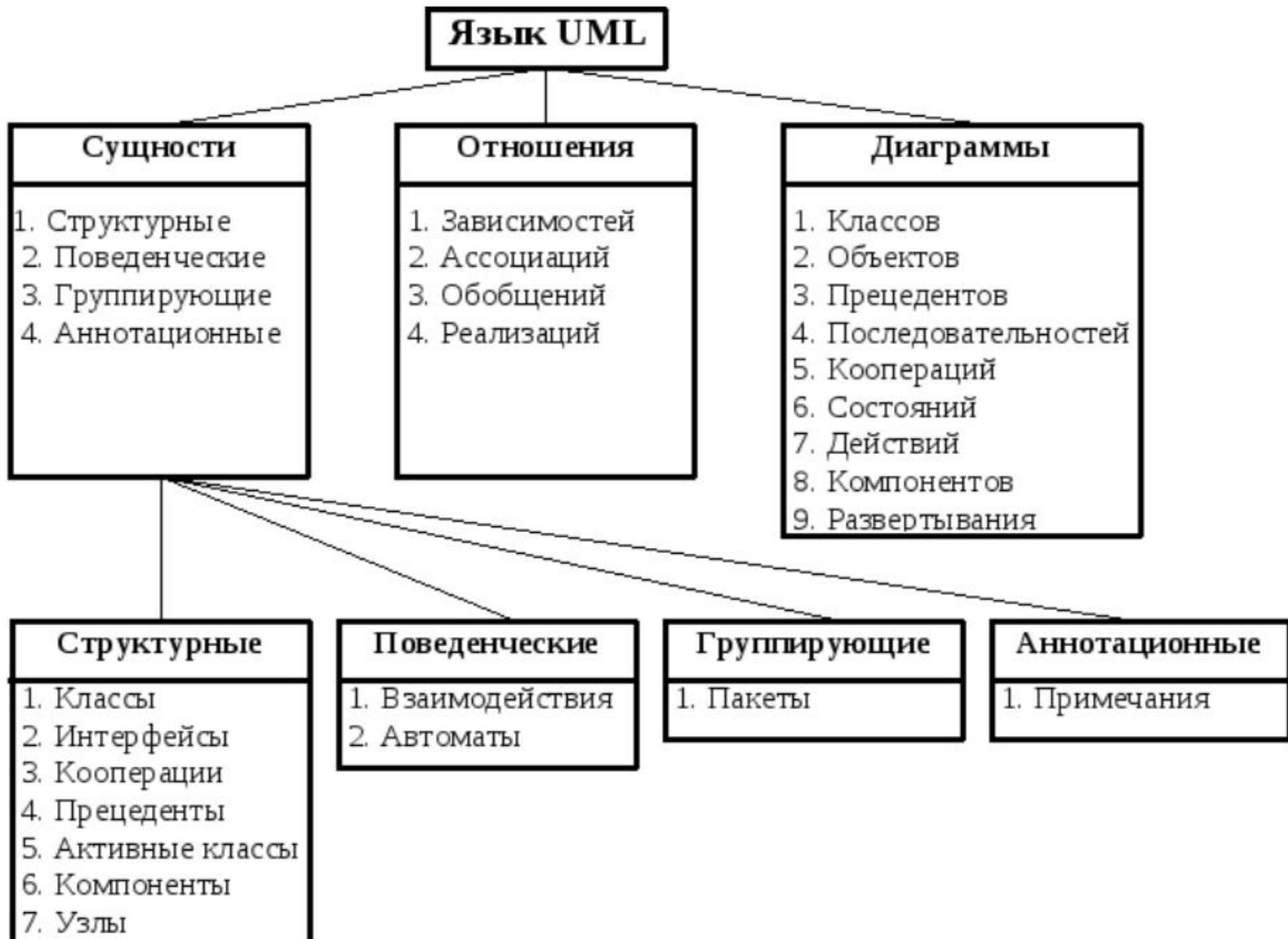
`position1 = position2`

`position3 = position2 + origin`

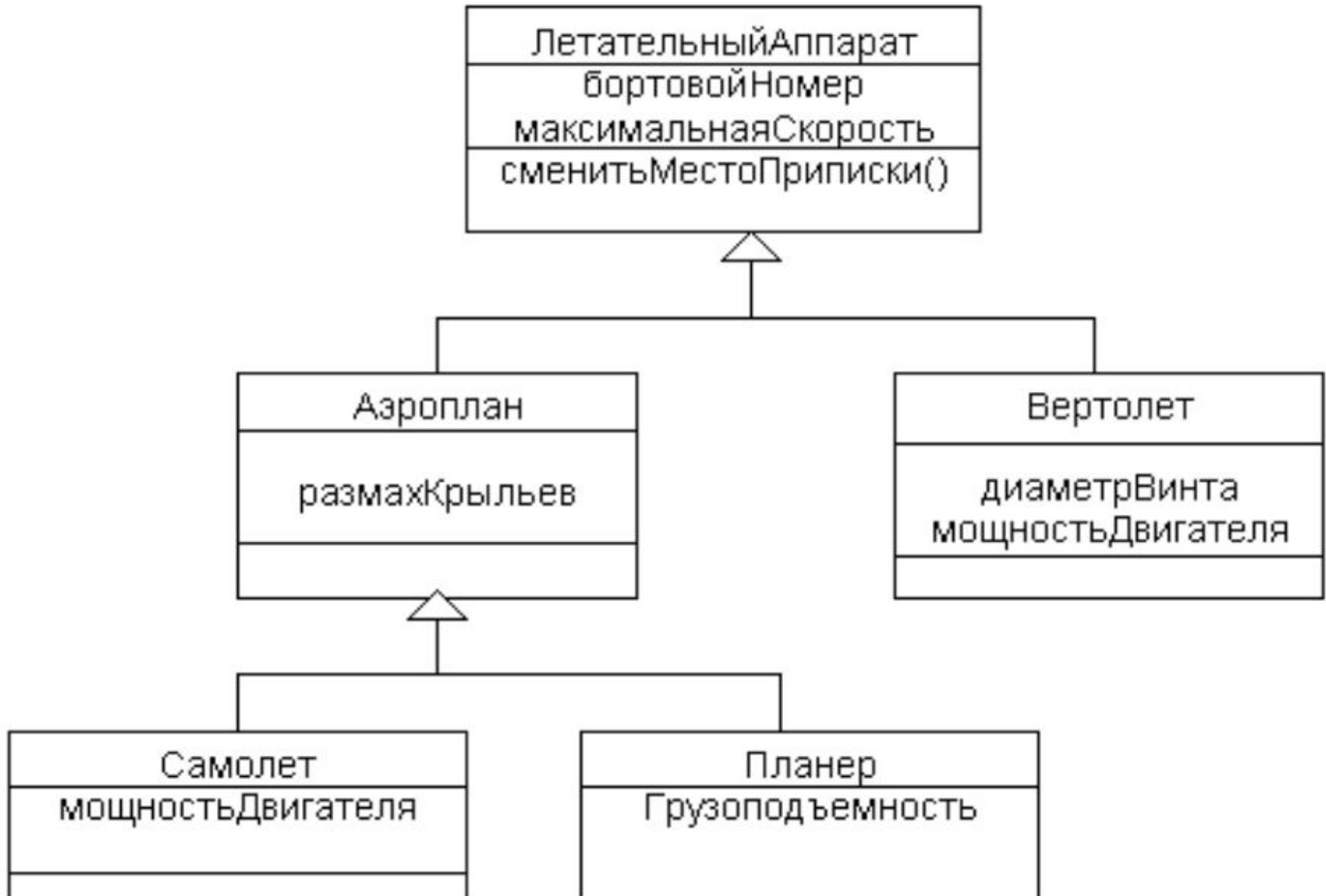
`isLager = position1 > position2`

Универсальный язык моделирования

UML



Универсальный язык моделирования (UML)



Универсальный язык моделирования (UML)

