



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Лекция №4
Объектно-ориентированная парадигма проектирования ИАС

Методы и средства проектирования информационно-аналитических систем

	<i>(наименование дисциплины (модуля) в соответствии с учебным планом)</i>
Уровень	специалитет
Форма обучения	<i>(бакалавриат, магистратура, специалитет)</i> очная
Направление(-я) подготовки	<i>(очная, очно-заочная, заочная)</i> 10.05.04 «Информационно-аналитические системы безопасности»
	<i>(код(-ы) и наименование(-я))</i>
Институт	Институт кибербезопасности и цифровых технологий (ИКБ)
Кафедра	<i>(полное и краткое наименование)</i> Информационно-аналитические системы кибербезопасности (КБ-2)
	<i>(полное и краткое наименование кафедры, реализующей дисциплину (модуль))</i>
Используются в данной редакции с учебного года	2023/24
	<i>(учебный год цифрами)</i>
Проверено и согласовано «___» _____ 20__ г.	<i>(подпись директора Института/Филиала с расшифровкой)</i>

Москва 2024 г.

Основные преимущества объектно-ориентированных методологий. Недостатки объектно-ориентированных методологий.

Учебные вопросы:

1. Сущность объектно-ориентированного подхода.
2. Базовые принципы объектно-ориентированного подхода.
3. Преимущества и недостатки объектно-ориентированного подхода.
4. Сравнение существующих методик.
5. Синтетическая методика.

1 Сущность объектно-ориентированного подхода

Первые объектно-ориентированные методы появились в начале 80-х гг. XX в., однако основные результаты были получены спустя 10 лет, когда Айвар Джекобсон, Джим Рембо и Грейди Буч представили свои подходы – Objectory, OMT (Object Modeling Technique – техника объектного моделирования) и Booch соответственно. Эти три метода были впоследствии объединены в один, из которого вырос унифицированный язык моделирования – UML (Unified Modeling Language).

Суть объектного подхода заключается в объектной декомпозиции, т.е. система представляется в виде совокупности объектов, которые в процессе взаимодействия обмениваются сообщениями. Дадим определение объекта.

Объект – это самостоятельная, самодостаточная сущность, обладающая состоянием, поведением и семантикой.

Модель процесса. Для объектно-ориентированного подхода характерен спиральный жизненный цикл системы. В отличие от структурного подхода, объектный подход не использует иерархических структур, в которых определение родителя не имеет смысла без определения всех его потомков. Вместо этого применяется система объектов, каждый из которых можно анализировать, проектировать, реализовывать и тестировать практически независимо от других. Таким образом, появляется возможность формирования итераций, на каждой из которых выполняются все основные этапы разработки для относительно небольшой группы объектов.

Объектно-ориентированный подход к созданию информационных систем позволяет объединить различные этапы разработки ИС: на первых этапах жизненного цикла ИС применяются методологии объектно-ориентированного анализа и проектирования, а затем – средства объектно-ориентированного программирования.

Целью анализа и проектирования является построение моделей разрабатываемой информационной системы. Определим особенности, присущие методам объектно-ориентированного анализа и проектирования (ООАП), сравнением с методами структурного анализа и проектирования.

2 Базовые принципы объектно-ориентированного подхода

Ключевым этапом при разработке любой информационной системы является системный анализ. Основной его принцип – декомпозиция. Объектно-ориентированная технология позволяет подойти к анализу сложных систем с точки зрения иерархических структур данных или иерархий объектов со свойствами инкапсуляции, наследования и полиморфизма. Использование этой технологии привело к появлению объектно-

ориентированного системного анализа, схемы которого отражают не только данные и их взаимосвязи, но и методы обработки данных. Этот подход позволяет получить наиболее естественную формализацию системы в целом.

Принципы объектно-ориентированного подхода. Объектно-ориентированному подходу присуще большинство принципов структурного подхода, а именно: «разделяй и властвуй», абстрагирование, формализация и непротиворечивость. Кроме того, выделяются следующие специфичные для него принципы:

1. **Уникальность.** Каждый объект уникален. Это самый базовый принцип, о котором многие даже не задумываются. Он означает, что никакие два даже абсолютно идентичных объекта нельзя считать одним и тем же объектом. Например, математика не различает два числа «5», однако в рамках объектно-ориентированного подхода два объекта, каждый из которых представляет число «5», будут различными. В терминах языков программирования это означает, что каждый объект имеет свой адрес в памяти, и две объектные переменные считаются равными только в том случае, если они указывают на один и тот же адрес.

2. **Классификация.** Все объекты объединяются в классы по принципу сходства структуры, поведения и семантики. Определение класса будет дано ниже. Здесь лишь заметим, что объект первичен по отношению к классу. Именно объекты объединяются в классы, формируют их. Ошибочно считать, что если в программе сначала описываются классы, а потом создаются их экземпляры, то класс первичен. Класс появляется только после анализа множества необходимых экземпляров.

3. **Инкапсуляция.** Есть два понятия инкапсуляции. В узком смысле инкапсуляция – это совместное «хранение» данных и методов их обработки. Это лишь частное свойство, которое нередко ошибочно считается единственным. В широком смысле инкапсуляция – это скрытие реализации за интерфейсом, т.е. объект обладает внутренней, известной лишь ему структурой и интерфейсом, через который другие объекты с ним взаимодействуют. Одним из важнейших следствий инкапсуляции является требование того, чтобы доступ к полям класса осуществлялся только через его методы. Разделение реализации и интерфейса позволяет менять реализацию, не влияя на способ взаимодействия с объектом.

4. **Наследование.** Наследование означает, что классы могут объединяться в иерархии наследования. Структура, поведение и семантика объектов наследуются вниз по иерархии. Наверху находятся более общие классы, внизу – более специфичные. Для наследования действует «правило подстановки». Суть его сводится к тому, что с объектами классов-потомков (специфичных) можно работать через интерфейс класса-родителя (общего), т.е. вместо объектов родителя можно подставлять объекты любого потомка.

5. **Полиморфизм.** Данный принцип означает возможность доступа к нескольким реализациям через один интерфейс. Способом реализации полиморфизма в объектно-ориентированных языках программирования является механизм переопределения унаследованных методов.

Объектно-ориентированная методика

Принципиальное отличие между функциональным и объектным подходом заключается в способе декомпозиции системы. Объектно-ориентированный подход использует объектную декомпозицию, при этом статическая структура описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами. Целью методики является построение бизнес-

модели организации, позволяющей перейти от модели сценариев использования к модели, определяющей отдельные объекты, участвующие в реализации бизнес-функций.

Концептуальной основой объектно-ориентированного подхода (по другим источникам) является объектная модель, которая строится с учетом следующих принципов:

- абстрагирование;
- инкапсуляция;
- модульность;
- иерархия;
- типизация;
- параллелизм;
- устойчивость.

Основными понятиями объектно-ориентированного подхода являются объект и класс.

Объект — предмет или явление, имеющее четко определенное поведение и обладающие состоянием, поведением и индивидуальностью. Структура и поведение схожих объектов определяют общий для них класс.

Класс — это множество объектов, связанных общностью структуры и поведения.

Следующую группу важных понятий объектного подхода составляют **наследование** и **полиморфизм**. Понятие полиморфизм может быть интерпретировано как способность класса принадлежать более чем одному типу. Наследование означает построение новых классов, на основе существующих с возможностью добавления или переопределения данных и методов.

Важным качеством объектного подхода является согласованность **моделей деятельности организации** и **моделей проектируемой информационной системы** от стадии формирования требований до стадии реализации. По объектным моделям может быть прослежено отображение реальных сущностей моделируемой предметной области (организации) в объекты и классы информационной системы.

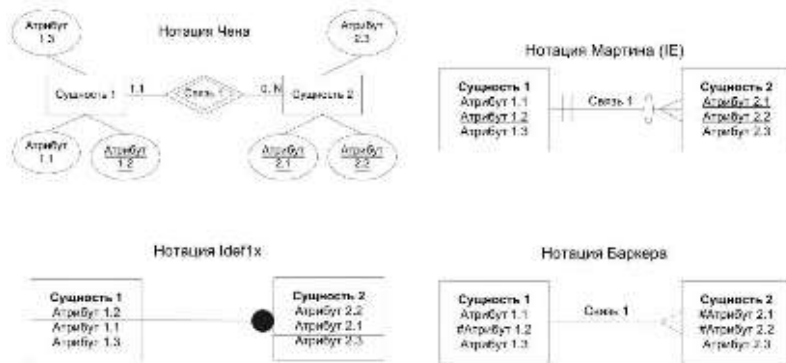
Большинство существующих методов объектно-ориентированного подхода включают **язык моделирования** и **описание процесса моделирования**. Процесс — это описание шагов, которые необходимо выполнить при разработке проекта. В качестве языка моделирования объектного подхода используется унифицированный язык моделирования UML, который содержит стандартный набор диаграмм для моделирования.

Диаграмма (Diagram) — это графическое представление множества элементов. Чаще всего она изображается в виде связного графа с вершинами (сущностями) и ребрами (отношениями) и представляет собой некоторую проекцию системы.

При использовании методологий структурного анализа и проектирования для построения модели информационной системы строятся различные типы структурных диаграмм, в частности, DF- и ER-диаграммы и т.д.



Пример DFD диаграммы

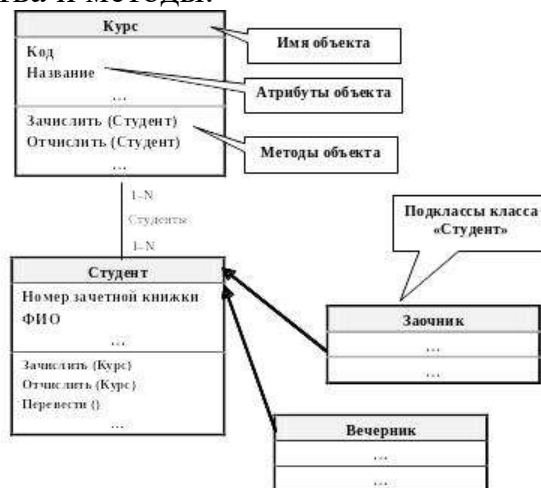


Примеры ER диаграмм в различных нотациях

Сущности, определяемые в ER-модели, можно назвать и классами объектов. Классы объектов определяются атрибутами (как и сущности), а также методами. Объекты обмениваются сообщениями. Таким образом, при использовании объектно-ориентированного подхода необходимо определить не только атрибуты (свойства) объектов, но и методы, оперирующие с классами объектов, и сообщения, которыми обмениваются объекты.

В объектной модели строятся диаграммы типов или классов. Можно выделить следующие основные отличия между моделью «сущность-отношение» и соответствующей моделью типов (классов):

- диаграмма типов не просто описывает данные, но и отображает методы, которые инкапсулированы в эти данные;
- типы данных в объектной модели разнообразнее и сложнее, чем в модели ER;
- классы могут включаться в другие классы;
- подклассы наследуют свойства и методы родительских классов и имеют свои свойства и методы.



Пример диаграммы типа (классов)

Для класса «Курс» определены атрибуты («Код», «Название» и др.) и методы, позволяющие зачислить конкретного студента на данный курс, отчислить его и т.д. На каждом курсе учится множество студентов, что выражено отношением между соответствующими объектами (один–ко–многим).

Класс «Студент» имеет подклассы «Вечерник» и «Заочник», наследующие его свойства и методы.

При ООАП используются следующие механизмы моделирования данных:

- **классификация** (объекты, обладающие одинаковыми свойствами и поведением, рассматриваются как члены одного класса);

- **подклассы и суперклассы** (объекты некоторого класса (подкласса) могут образовывать подмножество другого класса (суперкласса) и подклассы наследуют атрибуты и поведение своих суперклассов);

- **наследование** атрибутов и поведения (позволяет построить иерархию классов – реализуется концепция обобщение/конкретизация);

- **агрегирование** (позволяет создать сложные объекты из объектов-компонентов, определять отношения типа «часть-целое»).

Используя методы структурного анализа и проектирования, разработчик строит целый набор диаграмм (или таблиц, матриц, графов) различных типов (DFD, ERD и т.д.). Ему, как правило, нужно затратить значительные усилия на сборку на их основе готовых программных продуктов, так как используемые средства позволяют описать конкретные «срезы» исследуемой предметной области, но не дают возможность получить естественное формальное представление системы в целом.

Большинство методов ООАП включает как язык моделирования, так и описание процесса моделирования.

Язык моделирования – это нотация (в основном графическая), которая используется для описания проектов. Нотация представляет собой совокупность графических объектов, которые используются в моделях, она определяет синтаксис языка моделирования.

Унифицированный язык моделирования UML (Unified Modeling Language) является прямым объединением и унификацией созданных ранее методов и дополняет их новыми возможностями. Это язык для определения, представления, проектирования и документирования информационных систем. Он включает стандартный набор диаграмм и нотаций для моделирования:

- диаграммы вариантов использования (use case diagrams) – для моделирования бизнес-процессов организации (требований к системе);

- диаграммы классов (class diagrams) – для моделирования статической структуры классов системы и связей между ними;

- диаграммы поведения системы (behavior diagrams);

- диаграммы взаимодействия (interaction diagrams) – для моделирования процесса обмена сообщениями между объектами;

- диаграммы состояний (statechart diagrams) – для моделирования поведения объектов системы при переходе из одного состояния в другое;

- диаграммы деятельности (activity diagrams) – для моделирования поведения системы в рамках различных вариантов использования или моделирования деятельности;

- диаграммы реализации (implementation diagrams);

- диаграммы компонентов (component diagrams) – для моделирования иерархии компонентов (подсистем) системы;

– диаграммы размещения (deployment diagrams) – для моделирования физической архитектуры системы.

В ООАП технология объектов охватывает все стадии создания информационных систем. Объекты и перечисленные выше механизмы моделирования используются разработчиками вплоть до этапа конкретной реализации модели данных во внешней памяти. Средства анализа являются и средствами реализации. Это снимает барьер между различными этапами разработки информационной системы.

Кроме того, независимость сложных объектов позволяет не только повысить надежность системы, но и упрощает ее сопровождение, интеграцию с другими системами, что очень важно с точки зрения обеспечения живучести системы, ее развития.

Паттерн – это допускающие многократное использование оптимизированные решения проблем проектирования, с которыми мы сталкиваемся каждый день. Вопреки расхожему мнению, паттерны не являются готовыми алгоритмами, помогающими справиться с проблемой, а лишь описывают подход на верхнем уровне. Реализация идеи паттерна в каждом конкретном случае может сильно отличаться, даже если условия его применения были одинаковые.

Чем могут помочь паттерны ООАП:

- Лучше понимать существующее решение, спроектированное с использованием объектно-ориентированного подхода.
- Составлять проектные решения с оптимальной трудоемкостью, как для аналитика, так и для разработчика.
- Учесть возможность масштабирования системы при проектировании за счет использования подходящих для этого паттернов.
- Улучшить понимание решения на этапе передачи проекта команде технической поддержки.

Один из вариантов классификации паттернов – классификация по их сложности:

- Элементарные – базовые паттерны проектирования, которые являются основой для более сложных составных паттернов.
- Промежуточные – паттерны, состоящие из двух-трех элементарных паттернов.
- Паттерны «Банды четырех» - паттерны, описанные в книге Э. Гаммы, Р. Хелма, Р. Джонсона и Д. Влиссидеса. «Паттерны объектно-ориентированного проектирования». В ней представлены уже достаточно сложные по структуре паттерны, перед рассмотрением которых лучше ознакомиться со всеми элементарными паттернами и вариациями промежуточных.

3 Преимущества и недостатки объектно-ориентированного подхода

Достоинства объектно-ориентированного подхода. Основным достоинством является **устойчивость к изменениям и возможность повторного использования решений**. Эти свойства подхода вытекают из того, что вместо жестко связанных между собой функций используются относительно независимые объекты, чья реализация никому, кроме них самих, неизвестна. Соответственно **изменения получаются локальными** и есть возможность использовать готовые объекты в другом контексте.

Объектно-ориентированный подход обладает следующими **преимуществами**:

Объектная декомпозиция дает возможность создавать модели меньшего размера путем использования общих механизмов, обеспечивающих необходимую экономию выразительных средств. Использование объектного подхода существенно повышает

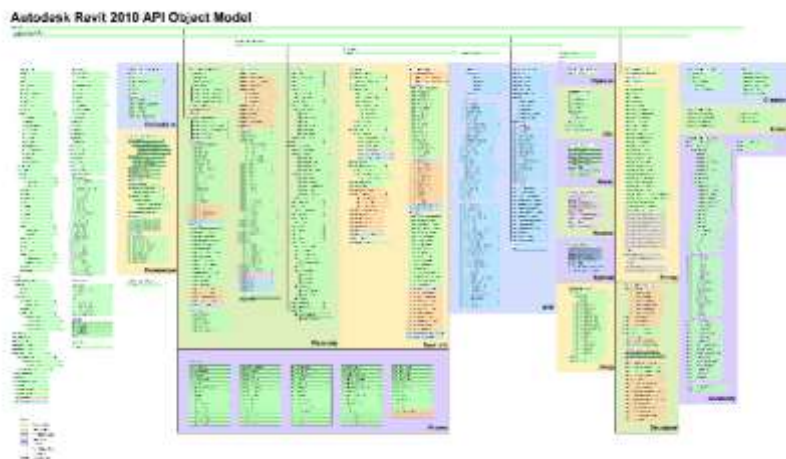
уровень унификации разработки и пригодность для повторного использования, что ведет к созданию среды разработки и переходу к **сборочному** созданию моделей.

Объектная декомпозиция позволяет избежать создания сложных моделей, так как она предполагает эволюционный путь развития модели на базе относительно небольших подсистем.

Объектная модель естественна, поскольку ориентирована на человеческое восприятие мира.

Недостатки объектно-ориентированного подхода. По сравнению со структурным подходом, объектно-ориентированный подход **проигрывает в производительности**. Основными причинами потери производительности являются инкапсуляция и полиморфизм. Действительно, для доступа к полям через методы и для поиска нужного метода по таблице виртуальных методов требуется процессорное время.

К **недостаткам** объектно-ориентированного подхода относятся высокие начальные затраты. Этот подход не дает немедленной отдачи. Эффект от его применения сказывается после разработки двух–трех проектов и накопления повторно используемых компонентов. Диаграммы, отражающие специфику объектного подхода, менее наглядны.



Пример представления объектной модели

4 Сравнение существующих методик

В функциональных моделях (DFD-диаграммах потоков данных, SADT-диаграммах) главными структурными компонентами являются функции (операции, действия, работы), которые на диаграммах связываются между собой потоками объектов.

Несомненным достоинством функциональных моделей является реализация структурного подхода к проектированию ИС по принципу "сверху-вниз", когда каждый функциональный блок может быть декомпозирован на множество подфункций и т.д., выполняя, таким образом, модульное проектирование ИС. Для функциональных моделей характерны процедурная строгость декомпозиции ИС и наглядность представления.

При функциональном подходе объектные модели данных в виде ER-диаграмм "объект - свойство - связь" разрабатываются отдельно. Для проверки корректности моделирования предметной области между функциональными и объектными моделями устанавливаются взаимно однозначные связи.

Главный **недостаток** функциональных моделей заключается в том, что процессы и данные существуют отдельно друг от друга — помимо функциональной декомпозиции существует структура данных, находящаяся на втором плане. Кроме того, не ясны условия выполнения процессов обработки информации, которые динамически могут изменяться.

Перечисленные недостатки функциональных моделей снимаются в объектно-ориентированных моделях, в которых главным структурообразующим компонентом выступает класс объектов с набором функций, которые могут обращаться к атрибутам этого класса.

Для классов объектов характерна иерархия обобщения, позволяющая осуществлять наследование не только атрибутов (свойств) объектов от вышестоящего класса объектов к нижестоящему классу, но и функций (методов).

В случае наследования функций можно абстрагироваться от конкретной реализации процедур (абстрактные типы данных), которые отличаются для определенных подклассов ситуаций. Это дает возможность обращаться к подобным программным модулям по общим именам (полиморфизм) и осуществлять повторное использование программного кода при модификации программного обеспечения. Таким образом, адаптивность объектно-ориентированных систем к изменению предметной области по сравнению с функциональным подходом значительно выше.

При объектно-ориентированном подходе изменяется и принцип проектирования ИС. Сначала выделяются классы объектов, а далее в зависимости от возможных состояний объектов (жизненного цикла объектов) определяются методы обработки (функциональные процедуры), что обеспечивает наилучшую реализацию динамического поведения информационной системы.

Для объектно-ориентированного подхода разработаны графические методы моделирования предметной области, обобщенные в языке унифицированного моделирования UML. Однако по наглядности представления модели пользователю-заказчику объектно-ориентированные модели явно **уступают** функциональным моделям.

При выборе методики моделирования предметной области обычно в качестве критерия выступает степень ее динамичности. Для более регламентированных задач больше подходят функциональные модели, для более адаптивных бизнес-процессов (управления рабочими потоками, реализации динамических запросов к информационным хранилищам) — объектно-ориентированные модели. Однако в рамках одной и той же ИС для различных классов задач могут требоваться различные виды моделей, описывающих одну и ту же проблемную область. В таком случае должны использоваться комбинированные модели предметной области.

5 Синтетическая методика

Как можно видеть из представленного обзора, каждая из рассмотренных методик позволяет решить задачу построения формального описания рабочих процедур исследуемой системы. Все методики позволяют построить модель **"как есть"** и **"как должно быть"**. С другой стороны, каждая из этих методик обладает существенными недостатками. Их можно суммировать следующим образом: недостатки применения отдельной методики лежат не в области описания реальных процессов, а в неполноте методического подхода.

Функциональные методики в целом лучше дают представление о существующих функциях в организации, о методах их реализации, причем чем выше степень детализации исследуемого процесса, тем лучше они позволяют описать систему. Под лучшим описанием в данном случае понимается наименьшая ошибка при попытке по полученной модели предсказать поведение реальной системы. На уровне отдельных рабочих процедур

их описание практически однозначно совпадает с фактической реализацией в потоке работ.

На уровне общего описания системы функциональные методики допускают значительную степень произвола в выборе общих интерфейсов системы, ее механизмов и т.д., то есть в определении границ системы. Хорошо описать систему на этом уровне позволяет **объектный подход**, основанный на понятии сценария использования. Ключевым является понятие о сценарии использования как о сеансе взаимодействия действующего лица с системой, в результате которого действующее лицо получает нечто, имеющее для него ценность. Использование критерия ценности для пользователя дает возможность отбросить не имеющие значения детали потоков работ и сосредоточиться на тех функциях системы, которые оправдывают ее существование. Однако и в этом случае задача определения границ системы, выделения внешних пользователей является сложной.

Технология потоков данных, исторически возникшая первой, легко решает проблему границ системы, поскольку позволяет за счет анализа информационных потоков выделить внешние сущности и определить основной внутренний процесс. Однако отсутствие выделенных управляющих процессов, потоков и событийной ориентированности не позволяет предложить эту методику в качестве единственной.

Наилучшим способом преодоления недостатков рассмотренных методик является формирование синтетической методики, объединяющей различные этапы отдельных методик. При этом из каждой методики необходимо взять часть методологии, наиболее полно и формально изложенную, и обеспечить возможность обмена результатами на различных этапах применения **синергетической методики**. В бизнес-моделировании неявным образом идет формирование подобной синергетической методики.

Идея синтетической методики заключается в последовательном применении функционального и объектного подхода с учетом возможности реинжиниринга существующей ситуации.

Рассмотрим применение синтетической методики на примере разработки административного регламента.

При построении административных регламентов выделяются следующие стадии:

Определение границ системы. На этой стадии при помощи анализа потоков данных выделяют внешние сущности и собственно моделируемую систему.

Выделение сценариев использования системы. На этой стадии при помощи критерия полезности строят для каждой внешней сущности набор сценариев использования системы.

Добавление системных сценариев использования. На этой стадии определяют сценарии, необходимые для реализации целей системы, отличных от целей пользователей.

Построение диаграммы активностей по сценариям использования. На этой стадии строят набор действий системы, приводящих к реализации сценариев использования;

Функциональная декомпозиция диаграмм активностей как контекстных диаграмм методики IDEF0.

Формальное описание отдельных функциональных активностей в виде административного регламента (с применением различных нотаций).

Вопросы для самоконтроля

1. В чем заключается сущность объектно-ориентированного подхода?

2. Чем объектно-ориентированный подход принципиально отличается от структурного подхода?
3. Что такое уникальность, классификация, инкапсуляция, наследование и полиморфизм?
4. Почему в рамках объектно-ориентированного подхода возможно использование спирального жизненного цикла?
5. На что тратится дополнительное процессорное время при использовании объектно-ориентированного подхода?
6. В чем преимущества объектно-ориентированного подхода перед структурным подходом?