Коллоквиум

Паттерны проектирования, ИСиТ-2

1. Перечислите основные парадигмы объектно-ориентированного программирования(ООП).

Инкапсуляция

Наследование

Полиморфизм

Абстракция

1. Поясните парадигму ООП «абстракция».

Абстракция: Процесс выделения общих характеристик объектов и игнорирования их специфических деталей. Абстракция позволяет создавать модели, которые описывают только существенные аспекты объектов, упрощая тем самым разработку и понимание программы.

Из лекции

15. Абстракция: моделирование системы, классы – типы объектов, объекты – экземпляры типов, классы/объекты отражают реальный объект только с точки зрения наблюдателя.

1. Поясните принцип «Бритва Оккама».

Бритва Оккама - не стоит множить сущее без необходимости

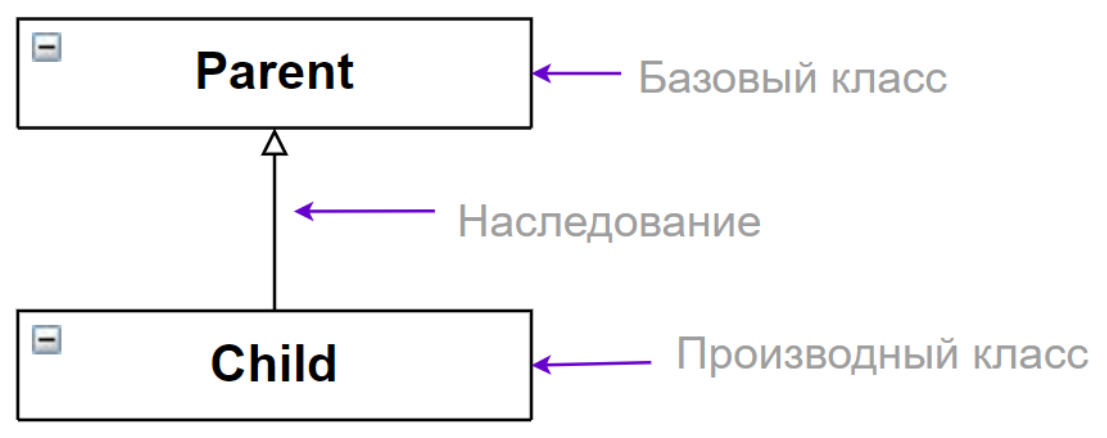
1. Поясните понятие ООП «интерфейс».

интерфейс - это контракт, определяющий набор методов и свойств, которые должен реализовать класс или объект. Интерфейс определяет, какие операции можно выполнять с объектом определенного типа, но не содержит реализации этих операций.

Из лекций

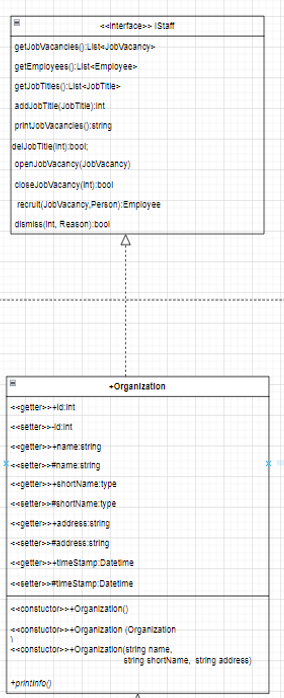
перечень сигнатур методов, предназначенных для реализации; контракт с фреймворком; спецификация фреймворка

1. Нарисуйте UML-диаграмму: класс A является производным от класса В.

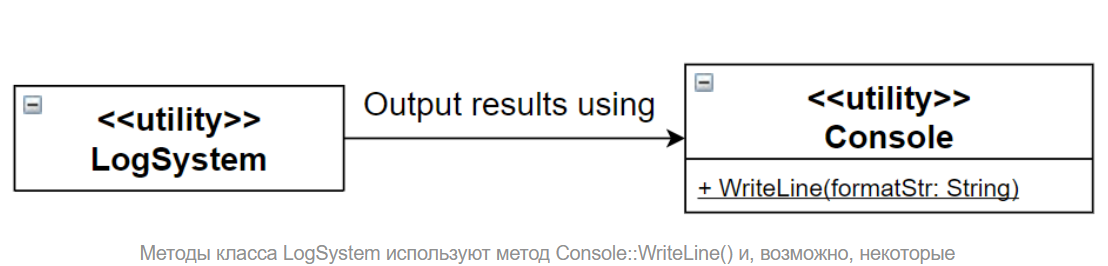


1. Нарисуйте UML-диаграмму: класс А реализует интерфейс I.





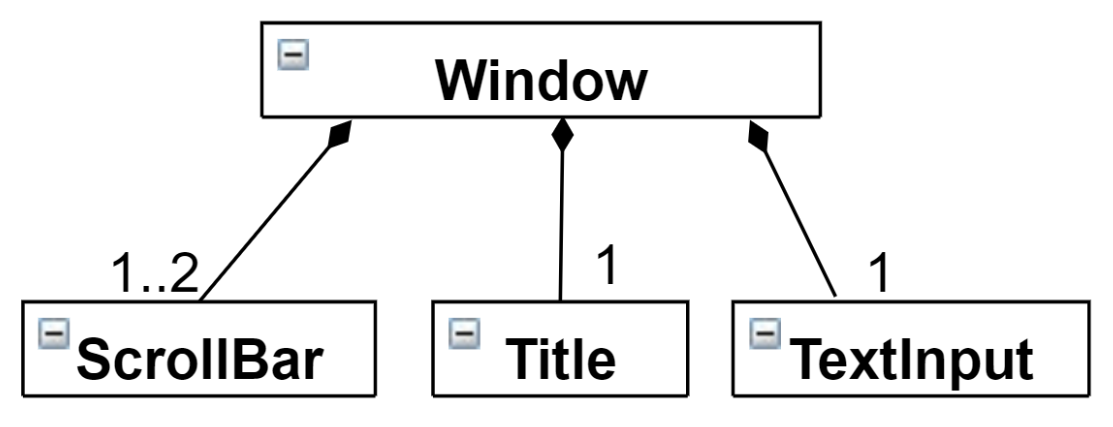
1. Нарисуйте UML-диаграмму: класс А использует тип класса B.



1. Нарисуйте UML-диаграмму: объект класса А содержит объект класса B.

Отношение композиции является частным случаем отношения агрегации. Однако у него есть одно отличие – классы-части, которые он соединяет с классом-целым, не могут существовать обособленно.

Понятное дело, что ни полоса прокрутки (ScrollBar), ни заголовок окна (Title), ни поле ввода команд (TextInput) не могут существовать отдельно от окна программы (Window).



1. Переведите с английского языка слово pattern.

узор, шаблон, схема

1. Дайте простое определение (максимум 4 слова) понятия паттерна.

Шаблон поведения или структуры.

1. Что такое «реализация паттерна»?

"Реализация паттерна" означает написание конкретного кода или программного решения в соответствии с принципами и идеями, предложенными определенным программистским паттерном

1. Поясните понятие «идиома».

устойчивый способ выражения некоторой составной конструкции

1. Поясните понятие «фреймворк».

это комплекс инструментов, библиотек, соглашений и шаблонов, предназначенных для упрощения разработки программного обеспечения. Он обеспечивает базовую структуру для создания и развертывания приложений, предоставляя готовые решения для типовых задач.

1. Поясните (расшифруйте) аббревиатуру SOLID.

SOLID - это акроним, который представляет собой пять основных принципов объектно-ориентированного программирования и дизайна

S - Принцип единственной ответственности. Каждый класс должен иметь только одну причину для изменения. Это означает, что класс должен быть ответственен только за одну часть функциональности программы.

O - Принцип открытости/закрытости Программные сущности (классы, модули, функции и т. д.) должны быть открыты для расширения, но закрыты для изменения. Это означает, что поведение программы должно быть легко расширяемым без необходимости изменения существующего кода.

L - Принцип подстановки Барбары Лисков. подклассы должны быть взаимозаменяемы с их суперклассами без нарушения свойств и предположений, сделанных о суперклассе

I - Принцип разделения интерфейса. Клиенты не должны зависеть от интерфейсов, которые они не используют.

D - Принцип инверсии зависимостей. Модули верхнего уровня не должны зависеть от модулей нижнего уровня. Оба типа модулей должны зависеть от абстракций.

1. Дайте простое определение (максимум 5 слов) SOLID.

Набор принципов качественного, масштабируемого кода

1. Поясните кратко Single Responsibility Principe (максимум 7 слов).

принцип единой ответственности

Каждый класс ответственен за один функционал

1. Поясните кратко Open/Close Principe (максимум 7 слов).

классы должны быть расширяемы (наследование), но неизменяемы

1. Поясните кратко Liskov Substitution Principe (LSP) (максимум 7 слов).

Базовые классы должны быть заменяемы производными.

1. Поясните кратко 1 правило LSP.

типы параметров одноименных метода производного класса должны иметь такой же тип как у базового или более абстрактный;

1. Поясните кратко 2 правило LSP.

Методы производного класса должны возвращать тот же тип данных, что и их базовый класс, или его подтип.

1. Поясните кратко 3 правило LSP.

производный метод не должен выбрасывать исключения не свойственные базовому(можно выбрасывать производные);

1. Поясните кратко 4 правило LSP.

производный метод не должен ужесточать предусловия;

1. Поясните кратко 5 правило LSP.

метод не должен ослаблять постусловия;

1. Поясните кратко 6 правило LSP.

Производный класс расширяет функциональность без изменения основной.

1. Поясните кратко 7 правило LSP.

производный класс не должен изменять приватные поля базового класса

1. Поясните кратко Interface Segregation Principle (максимум 7 слов).

не должны быть интерфейсы слишком «толстые», интерфейс должен соответствовать принципу «Бритва Оккама» - «не стоит множить сущее без необходимости», можно создавать несколько интерфейсов; не должно быть реализаций интерфейсов с пустой реализацией.

1. Поясните кратко Dependency Inversion Principle максимум 7 слов).

инверсия зависимости, open/closed principle – с другой стороны, изменение базового класса не должно приводить к изменению производных

1. Перечислите наименование 3 группы паттернов. Укажите количество паттернов в каждой группе.

Порождающте(5), структурные(7), поведенческие т=паттерны(10)

1. Библиотека классов реализует Abstract Factory. Что библиотека предоставляет программисту-пользователю? Что разрабатывает программист-разработчик библиотеки.

Abstract Factory: порождающий паттерн для создания семейств связанных объектов.

Программисту-пользователю библиотека предоставляет абстрактные классы и интерфейсы, которые позволяют создавать семейства связанных объектов, не привязываясь к их конкретным классам. Программист-разработчик библиотеки разрабатывает конкретные реализации абстрактных фабрик и продуктов, которые соответствуют определенным семействам объектов.

1. Библиотека классов реализует Facade. Что библиотека предоставляет программисту-пользователю? Что разрабатывает программист-разработчик библиотеки?

Программисту-пользователю библиотека предоставляет простой интерфейс к сложной системе классов или библиотеке. Программист-разработчик библиотеки разрабатывает этот простой интерфейс, который скрывает сложность и детали реализации системы.

1. Библиотека классов реализует Decorator. Что библиотека предоставляет программисту-пользователю? Что разрабатывает программист-разработчик библиотеки?

Программисту-пользователю библиотека предоставляет возможность динамически оборачивать объекты с помощью различных оберток (декораторов) для добавления нового функционала. Программист-разработчик библиотеки разрабатывает общий интерфейс для декораторов и конкретные декораторы, которые расширяют функциональность объектов.

1. Библиотека классов реализует Observer. Что библиотека предоставляет программисту-пользователю? Что разрабатывает программист-разработчик библиотеки?

Программисту-пользователю библиотека предоставляет возможность реализации паттерна "Наблюдатель" (Observer), где есть один наблюдаемый объект и множество наблюдателей. При изменении наблюдаемого объекта автоматически оповещаются все наблюдатели. Программист-разработчик библиотеки разрабатывает механизмы для подписки и отписки наблюдателей, а также для уведомления об изменениях в наблюдаемом объекте.