

55 lines (41 loc) · 4.08 KB

Preview

Code

Blame

Raw



Инкапсуляция

Инкапсуляция - один из основных принципов ООП, который означает упаковку данных и методов, работающих с этими данными, в единый компонент, называемый объектом.

Основная идея инкапсуляции заключается в том, что объект подобен закрытой капсуле, которая содержит в себе (*инкапсулирует*) всё необходимые данные и методы для моделирования предметной области. Другие объекты вызывают методы нашего объекта и получают необходимый результат, и при этом, они могут ничего не знать о том, как наш объект реализован внутри.

Пример: в классе Main вы используете объект Scanner для получения числа от пользователя. Вы вызываете метод `nextInt()`, вы получаете введенное число. При этом, вам не нужно знать какую "магию" выполнил сканер, что бы получить это число. Ваш код вызывает `scanner.nextInt()` и получает результат!

Реализация инкапсуляции предполагает реализацию двух принципов:

- сокрытие данных (Data Hiding)
- сокрытие реализации (Implementation Hiding)

Соккрытие данных (Data Hiding): Данные объекта должны быть скрыты от прямого доступа извне класса. Обычно это достигается путем объявления переменных класса как закрытых (**private**) и предоставления открытых (**public**) методов доступа к этим данным (геттеров и сеттеров). Таким образом, только методы класса могут изменять его состояние, обеспечивая контроль над данными и предотвращая нежелательные изменения извне.

Соккрытие реализации (Implementation Hiding): Внутренняя реализация объекта должна быть скрыта от клиентского кода. Клиентский код должен зависеть только от публичного интерфейса объекта, а не от его внутренней реализации. Это позволяет изменять внутреннюю реализацию объекта без влияния на клиентский код.

Реализация в коде

1. поля класса, как правило, делаем **private**
2. любое получение\изменение значений полей класса осуществляется через методы

Геттеры (getters) и сеттеры (setters)

Как было сказано выше любое получение\изменение значений полей класса осуществляется через методы. Частным случаем таких методов являются так называемые "методы доступа" или геттеры (getters) и сеттеры (setters)

Геттеры (getters) - методы, как правило, стандартного вида, для получения значения поля

```
public String getName() {  
    return name;  
}
```



Сеттеры (setters) - методы, как правило, стандартного вида, для установки значения поля.

```
public void setName(String name) {  
    this.name = name;  
}
```

