Цикл for

Цикл for - это управляющая конструкция программного кода, которая нарушает линейность исполнения алгоритма и позволяет выполнить указанный код множество раз.

for(инициализация; логическое выражение (условие); шаг (итерация))

команда

**for**(**int** i = 0; i < 5; i++) {

System.out.println(i);

}

Объе́ктно-ориенти́рованное программи́рование (ООП) — это методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Основные принципы ООП: абстракция, инкапсуляция, наследование, полиморфизм.

**Абстракция** –  означает выделение значимой информации и исключение из рассмотрения незначимой. С точки зрения программирования это правильное разделение программы на объекты. Абстракция позволяет отобрать главные характеристики и опустить второстепенные.

Пример: описание должностей в компании. Здесь название должности значимая информация, а описание обязанностей у каждой должности это второстепенная информация. К примеру главной характеристикой для “директор” будет то, что это должность чем-то управляет, а чем именно (директор по персоналу, финансовый директор, исполнительный директор) это уже второстепенная информация.

**Инкапсуляция** – свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе. Для Java корректно будет говорить, что инкапсуляция это “сокрытие реализации”. Пример из жизни – пульт от телевизора. Мы нажимаем кнопочку “увеличить громкость” и она увеличивается, но в этот момент происходят десятки процессов, которые скрыты от нас. Для Java: можно создать класс с 10 методами, например вычисляющие площадь сложной фигуры, но сделать из них 9 private. 10й метод будет называться “вычислитьПлощадь()” и объявлен public, а в нем уже будут вызываться необходимые скрытые от пользователя методы. Именно его и будет вызывать пользователь.

**Наследование** – свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником, дочерним или производным классом.

Полиморфизм (polymorphism) (от греческого polymorphos) - это свойство, которое позволяет одно и то же имя использовать для решения двух или более схожих, но технически разных задач. Целью полиморфизма, применительно к объектно-ориентированному программированию, является использование одного имени для задания общих для класса действий. Выполнение каждого конкретного действия будет определяться типом данных.