1. ООП (Объектно-ориентированное программирование) – метод программирования, основанный на представлении программы в виде совокупности объектов, образующих иерархию.

Объект ООП – сущность или экземпляр класса, способный сохранять своё состояние (Информацию) обеспечивающиеся набор операций (поведение) для проверки и изменения этого состояния

Наследование ООП – механизм, при помощи которого можно создавать классы внутри других классов с частично или полностью идентичной функциональностью

Как запретить наследование класса:

При объявлении класса необходимо написать слово final (Можно также использовать для полей и модификаторов), в таком случае никто не может унаследовать его элементы

Абстракция ООП – выделение наиболее важных характеристик внутри объекта

Инкапсуляция ООП – базовый принцип ООП, при котором атрибуты и поведение объекта объединяются в одном классе, при чём внутренняя реализация скрывается от пользователя.

Основная идея инкапсуляции заключается в том, чтобы скрыть внутренние процессы и детали реализации программы от пользователя и предоставить лишь необходимые ему инструменты для работы (То есть, упростить интерфейс).

Для реализации инкапсуляции можно использовать модификаторы доступа или геттеры и сеттеры

Полиморфизм ООП – свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта

Реализация полиморфизма

-Через переопределение методов (@Override) – реализование метода в классе-наследнике так, чтобы он имел идентичную структуру с методом класса-предка, но предоставлял иное поведение

-Через перегрузку методов – использование метода с одним и тем же именем, но с различным набором параметров, их типов

public class Program{

public static void main(String[] args) {

System.out.println(sum(2, 3)); // 5

System.out.println(sum(4.5, 3.2)); // 7.7

System.out.println(sum(4, 3, 7)); // 14

}

static int sum(int x, int y){

return x + y;

}

static double sum(double x, double y){

return x + y;

}

-Через коварианты возвращаемых типов – возвращение объекта и его потомков

-Через абстрактные классы – классы, в которых нельзя создать объект, они могут лишь предоставлять функции

Виды полиморфизма:

В статическом полиморфизме выбор определяется во время компиляции. В динамическом полиморфизме выбор определяется во время выполнения.

1. Класс – шаблон или описание какого-то объекта

Объект – экземпляр класса

<https://javarush.com/groups/posts/principy-oop>

https://habr.com/ru/post/87205/

1. Члены класса.

Модификаторы доступа: 1. Public – модифиактор, предоставляющий полный доступ к полям, методам, классам. Protected – поля и методы доступны классам только в пределах одинакового пакета или всем классам наследникам. Private – ограничивает видимость данных в пределах одного класса.

2. Static – указывает на то, что переменная является переменной класса, а не экземпляром метода класса

3. Final - запрещает присвоение значений вне тела класса, в котором она объявлена.

4. Создание и инициализация объектов

5. Public – модифиактор, предоставляющий полный доступ к полям, методам, классам. Protected = default – поля и методы доступны классам только в пределах одинакового пакета или всем классам наследникам. Private – ограничивает видимость данных в пределах одного класса.

Static – указывает на то, что переменная является переменной класса, а не экземпляром метода класса

Final - запрещает присвоение значения значений вне тела класса, в котором она объявлена.

6. final - запрещает присвоение значения значений вне тела класса, в котором она объявлена.

Static - указывает на то, что переменная является переменной класса, а не экземпляром метода класса

7. Пакет – механизм, позволяющий организовать Java классы в пространстве имен

import позволяет импортировать пакеты, модули и имена в пространство имен, в котором инструкция import выполняется

Записи (record) предоставляют лаконичный синтаксис для объявления классов, которые являются простыми носителями постоянных, неизменяемых наборов данных.

Появятся в Java 14 в виде preview feature.

record Person(String name, int age) { }

Запечатанные (sealed) типы — это классы или интерфейсы, которые накладывают ограничения на другие классы или интерфейсы, которые могут расширять или реализовывать их.

super - текущий экземпляр родительского класса

Загрузчик классов Java — это часть среды выполнения Java, которая динамически загружает классы Java в виртуальную машину Java.

БАЙТ КОД

New – создание переменной

Duv -