Задание 5

Зайдите на сайт Oracle. Используя поисковые механизмы Oracle, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

1. Oracle: Полиморфизм

<https://docs.oracle.com/javase/tutorial/java/IandI/polymorphism.html>

Словарное определение полиморфизма относится к принципу биологии, согласно которому организм или вид могут иметь множество различных форм или стадий. Этот принцип можно также применить к объектно-ориентированному программированию и таким языкам, как язык Java. Подклассы класса могут определять свое собственное уникальное поведение и при этом использовать некоторые из тех же функций родительского класса.

1. Oracle: Модификаторы доступа

<https://docs.oracle.com/javase/tutorial/java/javaOO/variables.htm>

Существует несколько видов переменных:

Переменные-члены в классе — они называются полями.

Переменные в методе или блоке кода — они называются локальными переменными.

Переменные в объявлениях методов — они называются параметрами.

В Java предоставляются следующие четыре модификатора доступа, определяющие границы области видимости компонентов программы.

1. Модификатор доступа private - ограничивает область видимости пределами класса.

2. Модификатор доступа рublic - не ограничивает область видимости.

3. Модификатор доступа protected - ограничивает область видимости пределами пакета и всеми подклассами.

4. Модификатор доступа по умолчанию - область видимости ограничивается пределами пакета.

1. Oracle: Наследование

<https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.htm>

Определение: Класс, производный от другого класса, называется подклассом (также производным классом, расширенным классом или дочерним классом). Класс, от которого происходит подкласс, называется суперклассом (также базовым классом или родительским классом).

За исключением Object, у которого нет суперкласса, каждый класс имеет один и только один прямой суперкласс (одиночное наследование). В отсутствие какого-либо другого явного суперкласса каждый класс неявно является подклассом Object.

Классы могут быть производными от классов, производных от классов, производных от классов и т. д., и в конечном счете производными от самого верхнего класса, Object. Говорят, что такой класс происходит от всех классов в цепочке наследования, восходящей к Object.

Идея наследования проста, но мощна: когда вы хотите создать новый класс, а уже есть класс, который включает часть кода, который вам нужен, вы можете получить новый класс из существующего класса. При этом вы можете повторно использовать поля и методы существующего класса без необходимости писать (и отлаживать) их самостоятельно.

Подкласс наследует все члены (поля, методы и вложенные классы) от своего суперкласса. Конструкторы не являются членами, поэтому они не наследуются подклассами, но конструктор суперкласса может быть вызван из подкласса.