1. **Introdução:**
   1. **Orientação a Objetos**

A programação orientada a objetos(POO) é um dos principais paradigmas da programação. A orientação a objetos permite que os códigos sejam encapsulados em “pacotes” – que são conhecidos como classes –, o que torna possível a abstração e também uma melhor reutilização do código. Os objetos, responsáveis pelo nome “Orientada a Objetos”, são instâncias de uma classe, que possui métodos e variáveis específicos, com um valor “único” para cada objeto. Com a utilização do POO é possível criar tipos mais complexos de objetos, quebrando a barreira “imposta” pelos tipos primitivos, além de tornar seu código mais estruturado. (Anotações de aula).

* 1. **Polimorfismo e Herança**

Quando uma classe estende outra, ela herda todos os métodos e atributos que estão na classe mãe (ou superclasse), organizando assim as classes de forma em que a superclasse é a classe mais geral e conforme existem suas “filhas” as classes passam a ser cada vez mais específicas.

É possível sobrescrever os métodos da classe mãe, fazendo assim com que cad­­­a classe tenha um comportamento diferente. Por exemplo, cães e gatos “estendem” mamíferos e fazem um som diferente, como se o método “fazSom()” tivesse sido sobrescrito em cada um deles.

Polimorfismo também pode ser exemplificado como dito acima. Quando se tem classes (cães e gatos) que herdam o mesmo método (fazSom()) de uma superclasse (no exemplo mamíferos) e esse método faz coisas diferentes para cada classe específica, à isso dá-se o nome de Polimorfismo. (Anotações de aula)

* 1. **Multiprogramação**

Embora pareça, o computador ainda não é capaz de fazer algo “ao mesmo tempo”, o que ocorre é o chaveamento dos processamentos no processador de forma tão rápida que parace que tudo está sendo executado ao mesmo tempo.

Tal ato foi muito perceptível no desenvolvimento do jogo, uma vez que era preciso remover um Asteroid da lista enquanto essa era percorrida para imprimir seus elementos. Ao tentar fazer isso o erro de Concurrent Modification aparecia, para resolver tal problema foi necessária a criação de seções críticas no código, onde, nessas seções, o processador seria impedido de chavear o processamento, ou seja, somente apaga o Asteroid, não chaveando assim para a parte do código que imprimia a lista enquanto o Asteroid era excluído. (Anotações de aula)