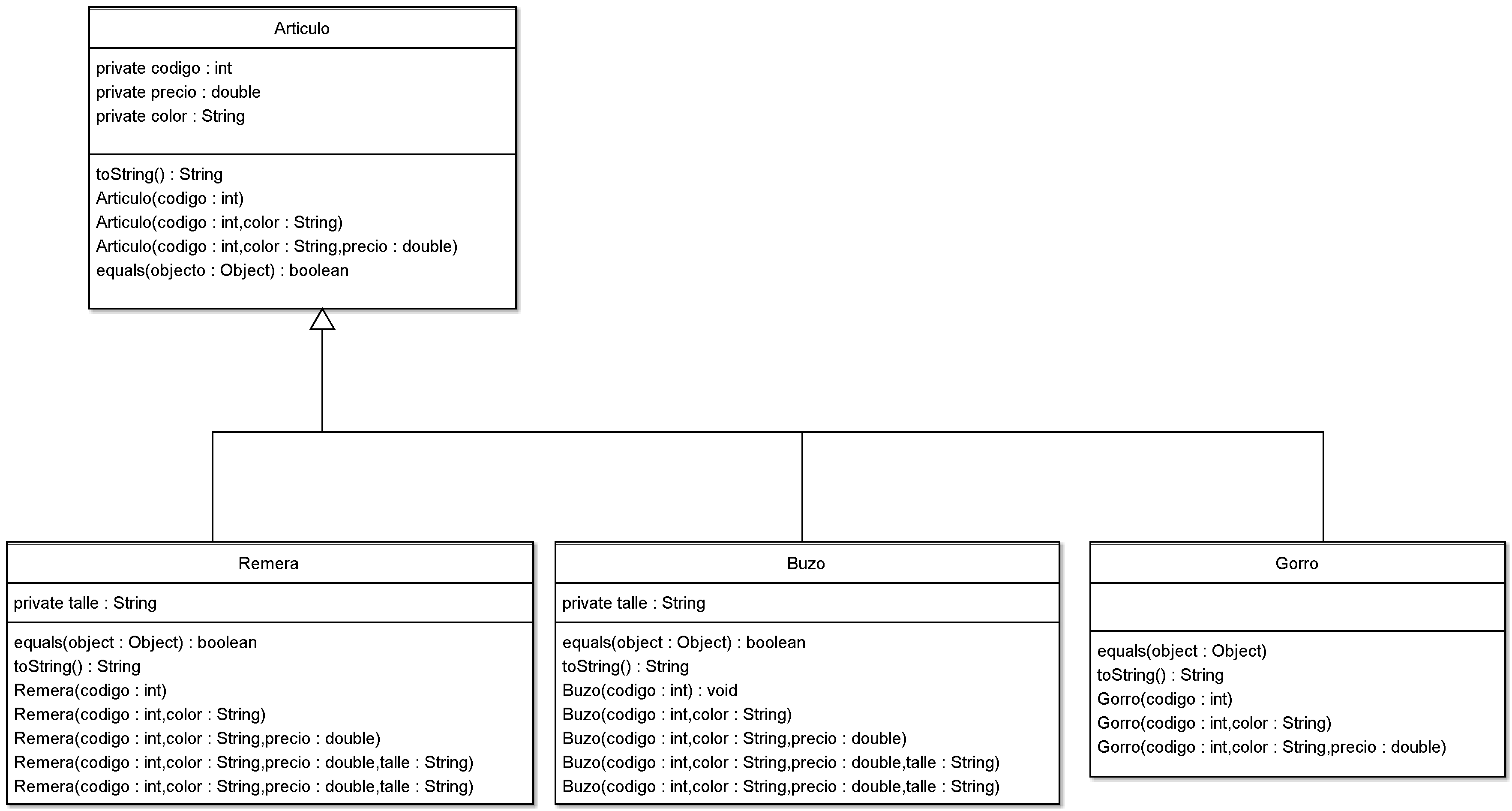
**Práctica “Encadenamiento de Constructores”**

**Temas:**

* Encadenamiento de constructores
* this() y super()
* equals() y toString()

Para realizar esta práctica cree un proyecto llamado *practica6* en su espacio de trabajo

1. Codifique las clases de acuerdo al diagrama de clases dado a continuación. Las clases deben estar ubicadas en el paquete **paqueteej1.**



1. Defina los métodos de acceso a las variables de instancia, proveyendo ocultamiento y encapsulamiento (setters y getters).
2. Los constructores de la clase **Articulo** con más de un argumento invocan a un constructor de su misma clase con un argumento menos que el que se está definiendo.
3. Los constructores de la clase **Remera** invocan a los constructores de la superclase con la misma cantidad y tipo de argumentos. El constructor con 4 argumentos invoca al constructor de la superclase con 3 argumentos e inicializa el talle con el valor que recibe en el argumento.
4. El constructor de la clase **Buzo** de un argumento invoca al constructor de la superclase con un solo argumento. Los constructores de Buzo con más de un argumento invocan a un constructor de la misma clase con un argumento menos.
5. Los constructores de la clase **Gorro** con más de un argumento invocan a los constructores de la misma clase con un argumento menos. El constructor con un solo argumento NO deberá invocar al de la superclase.
6. ¿Pudo compilar las clases? ¿Qué problemas surgieron y por qué? ¿Cómo los solucionó?
7. En cada clase de la jerarquía provea un método que compare si 2 objetos de la misma clase son iguales. ¿Qué método debería sobreescribir?
8. En cada clase de la jerarquía provea un método que imprima en pantalla los datos de los objetos de dicha clase. En las subclases de **Articulo** el método deberá imprime los datos con el siguiente estilo: Remera talle: XL código: 101 color: Rojo precio: $95: Gorro código: 23 color: Negro precio: $40.50; Buzo talle: S código: 45689 color: rayado precio: $87. ¿Qué método debería sobreescribir?
9. Escriba una clase llamada **TestArticulos** para testear las clases creadas. En su método main():

* Cree un objeto Remera usando el constructor con un argumento. Luego asígnele valor al resto de sus atributos usando los métodos seters.
* Cree un objeto Remera usando el constructor con 2 argumentos. Luego asígnele valor al precio y al talle usando los métodos setters.
* Cree un objeto Remera usando el constructor con 3 argumentos. Luego asígnele valor al talle con el método setter.
* Cree un objeto Remera usando el constructor con 4 argumentos.
* Repita la misma secuencia y cree 4 objetos Buzo y 3 objetos Gorro.
* Imprima en pantalla las 4 remeras, los 4 buzos y los 3 gorros.

1. Ahora comente los métodos toString() de las clases Remera, Buzo y Gorro y vuelva a compilar. Ejecute la clase **TestArticulos** ¿Qué diferencias encuentra con la versión original? ¿Porqué?
2. En el método main() cree un nuevo objeto Remera con los mismos valores de la cuarta remera que creó. Compárela usando el método equals(). ¿Qué resultado obtuvo? Compárales con el operador == ¿Obtuvo el mismo resultado?
3. Ahora, agregue la siguiente línea de código al método main():

**Remera remera6 = remera5; // remera 5 es la remera creada en el inciso anterior**

Compare la remera6 y la remera5, usando el método equals() y el operador ==. ¿Qué resultados obtuvo?