Renaldi Soeryadi

72190301

1. Apa yang dimaksud dengan Inheritance, jelaskan contohnya dengan C#

*Inheritance* merupakan sebuah kelas yang mampu mewariskan kemampuan yang dimiliki kepada kelas-kelas turunan (Kelas Induk / Superclass), yang merupakan kelas yang lebih spesifik terhadap suatu logika daripada kelas induknya.

Format penulisan untuk operasi Inheritance adalah sebagai berikut.

Class <nama kelas induk> { 🡨 Kelas induk / Superclass

<isikan kode-kode yang anda inginkan disini>;

}

Class <nama kelas turunan> : <nama kelas induk> { 🡨 Kelas Turunan

<isikan kode-kode yang anda inginkan disini>;

}

Dst.

Contohnya seperti berikut.

Text

Description automatically generated

Pada contoh di atas, kita ibaratkan kelas Barang merupakan kelas induk atau sebagai *Superclass*, sedangkan kelas Ukuran dan kelas Bentuk merupakan kelas turunan, yang ditandai dengan adanya tanda titik dua ( : ) setelah nama kelas yang disusul dengan nama kelas Induknya.

Nanti hasilnya akan tampak sebagai berikut.

A picture containing text, dark, orange

Description automatically generated

Rahasia mengapa kelas Ukuran mampu memanggil deskripsi yang ada dalam kelas Barang adalah sebagai berikut.

Seorang anak pasti akan memiliki beberapa sifat yang diturunkan dari kedua orang tuanya ketika dia lahir dan tumbuh berkembang di dunia. Menggunakan analogi anak dengan orang tua tersebut, maka kita tahu bahwa pasti bahwa kelas turunan pasti membawa semua informasi yang terkandung dalam kelas Utamanya atau kelas Induknya ketika kelas tersebut dipanggil menggunakan proses Instansiasi.



Proses Instansiasi

Nah, dalam contoh diatas, kelas Ukuran sudah memiliki semua informasi yang ada dalam kelas Barang berkat proses *Inheritance* tadi, dengan tanda : setelah nama kelas turunannya dan sebelum nama kelas induknya (Superclass).

Text

Description automatically generated

Maka ketika kelas Ukuran dipanggil menggunakan proses Instansiasi, maka praktis semua informasi yang ada dalam Kelas Barang (Superclass) dan Kelas Ukuran (Kelas Turunan) maka praktis digabung menjadi satu dalam Kelas Ukuran, sehingga Kelas Ukuran dapat memanggil semua informasi yang ada dalam kelas Barang.

1. Apa yang dimaksud dengan Polymorphism, jelaskan contohnya di C#

Polymorphism merupakan sebuah konsep Generalisasi yang mampu menurunkan suatu metode kepada sub-kelas nya dan melakukan tugas yang berbeda daripada kelas Induknya.

Contohnya seperti berikut.

Text

Description automatically generated

Nanti, hasilnya adalah sebagai berikut.

Text

Description automatically generated

Rahasia mengapa hasil yang dikeluarkan oleh system berbeda daripada yang didefinisikan sebelumnya adalah sebagai berikut.

Polimorfisme sebetulnya merupakan kelanjutan dari metode *Inheritance,* dimana pembeda antara metode *Inheritance* dengan Polimorfisme adalah metode *Inheritance* akan mewariskan semua informasi yang ada dalam kelas induk dan menyalin semua informasi yang ada sesuai dengan keinginan pengguna, sedangkan Polimorfisme akan mengubah dan mengesampingkan (*override*) informasi yang ada dalam kelas turunannya dan menyalin informasi yang ada dalam kelas Induknya.

Rahasia mengapa bisa begitu terletak pada proses Instansiasi yang ada dalam Kelas Utama (class Utama).

Text

Description automatically generated

Kelas induk (Superclass) yang merupakan kelas bergerak, bertindak sebagai Penimpa dengan memprioritaskan informasi yang ada dalam kelas Bergerak tersebt ketimbang kelas Turunannya (Anjing, Burung, Ikan) dengan cara:

<Kelas Induk> <nama Variabel> = new <nama kelas turunan>;

Dengan menyebutkan Kelas Induknya terlebih dahulu (di sebelah kiri) daripada nama Kelas Turunan yang akan dipanggil pada sebelah kanan, maka secara otomatis

Semua tugas akan beralih kepada kelas Induknya.

1. Sebutkan tipe2 collection yang ada di C#
2. Array
3. ArrayList
4. List<T>
5. Queue<T>
6. Stack<T>
7. Dictionary<T>
8. Jelaskan perbedaan Abstract Class dan Inteface pada C#

Perbedaan Abstract Class dan Interface dalam C# adalah sebagai berikut.

1. Abstract Class (Template Class) merupakan sebuah kelas yang hanya bertindak sebagai Kelas Pendefinisi, dimana dalam kelas tersebut semua informasi dan logika pemrograman tidak dapat dieksekusi dalam bentuk apapun dan hanya dapat diturunkan kedalam Kelas Turunan, dimana dalam Kelas Turunan tersebut semua informasi yang ada dalam Kelas Abstrak tersebut nantinya akan di-*Override* untuk dapat dieksekusikan.

Pada Abstract Class sendiri, kitab oleh memiliki metode atau property yang sudah ada acara implementasinya. Jadi, karena dalam semua informasi yang ada dalam kelas Abstrak tidak dapat dieksekusi dalam kelas tersebut, maka proses implementasinya harus ada dalam kelas turunannya.

1. Interface merupakan kelas yang dapat digunakan untuk mengimplementasikan (mengeksekusi) semua metode, property, dan logika pemrograman yang ada didalam kelas tersebut, dimana untuk dapat mengeksekusi semua metode dan property yang ada, system harus menyetujui semua persyaratan yang diperlukan dalam kelas tersebut.

Pada Interface, kita dapat secara langsung menetapkan semua property dan metode dan harus diimplementasikan dalam kelas yang sama dengan Interface tersebut.

1. Jelaskan perbedaan Metode *Overloading* dan Metode *Overriding* pada C#

Metode Overloading adalah metode yang memperbolehkan anda untuk mengidentifikasn suatu metode yang sama namun dengan persyaratan yang berbeda.

Sedangkan Metode Overriding adalah metode yang memperbolehkan anda untuk mendefinisikan suatu metode yang sama seperti metode Induknya, hanya saja nanti memiliki fungsi yang berbeda ketika dijalankan.