Polymorphism কি?

আমি সবসময়ই বইয়ের টিপিক্যাল সঙ্গাকে অপছন্দ করি। তাই আমি যা বুঝেছি সেটা আমার মতো করেই বলার চেষ্টা করবো। Polymorphism এর বাংলায় অর্থ দ্বারায় বহুরূপতা। কি, সাথে সাথেই আপনার চোখে মেয়েদের চেয়ারা ভেসে উঠেছে? মেয়েরা একেক ছেলের সাথে একেক রকম ব্যবহার বা একেক সময়ে একেক রকমের আচরন করে। আর হ্যা পলিমরফিজম কিছুটা একই রকম। অর্থাৎ একই লোকের অবস্থাভেদে আচরনের ভিন্নতার নামই পলিমরফিজম। তাহলে প্রশ্ন হলো প্রোগ্রামিং’এ পলিমরফিজম কি? প্রোগ্রামিং’এ পলিমরফিজম হলো প্যারেন্ট ক্লাসের মেথডসমূহের একেক চাইল্ড ক্লাসে একেকরকম আচরন করা। অর্থাৎ প্যারেন্ট ক্লাসে মেথডের একটি ব্লুপ্রিন্ট থাকবে আর চাইল্ড ক্লাস আর চাইল্ড ক্লাস সেই মেথডগুলোর তাদের নিজের মতো করে ক্যারেক্টারিস্টিক সেট করবে। মনে হচ্ছে বুঝতে বুঝতে একটু কষ্ট হচ্ছে। সমস্যা নেই মাথা উপর থেকে যাবার আগেই আমরা একটা বাস্তব জীবন থেকে উদাহরন দেইঃ  
ধরুন আপনার চাচার দুই ছেলে। একজন খুব ভদ্র আর একজন ইভটিজার। এখানে দুইজনই আপনার চাচার ছেলে, অর্থাৎ আপনার চাচা যদি ক্লাস হয় তবে ক্যারেক্টার হচ্ছে তার একটি মেথড। এখানে ছেলেরা হচ্ছে তার একেকটি চাইল্ড ক্লাস আর এই চাইল্ড ক্লাসগুলো আপনার চাচার ব্লুপ্রিন্ট থেকে নেয়া এবং তারা দুজেনই আপনার চাচার ছেলে কিন্তু তাদের বৈশিষ্ট্য বা চরিত্র ভিন্ন ভিন্ন।  
জানি এখনো মাথা উপর দিয়েই যাচ্ছে অনেকের। ধৈর্য ধরে পরতে আগাতে থাকুন আশা করি আস্তে আস্তে মাথার নিচে নামবে।

পলিমরফিজমের প্রধান একটি বৈশিষ্ট্য হলো এর common interface. অর্থাৎ সকল চাইল্ড ক্লাসের কমন বৈশিষ্ট্যগুলো প্যারেন্ট ক্লাসে শুধু উল্লেখ করা থাকবে। চাইল্ড ক্লাসগুলো শুধু প্যানেন্ট ক্লাসের ব্লুপ্রিন্ট থেকে নিজেকে তৈরী করবে। Polymorphism কে দুটি উপায়ে বর্ননা করা যায় এবং এদের প্রত্যেকেরই নিজস্ব ব্যবহার আছে।

1. Interface
2. Abstract Classes

Interface

Polymorphism এর একটি বহুল ব্যবহৃত অংশ হলো Interface. এটি OOP এর খুবই গুরুত্বপূর্ন একটি ফিচার। ইন্টারফেস বোঝার পূ্র্বে আমরা একটি উদাহরন দেখি, যারা ওয়েব ডেভলপমেন্টে কাজ করেন তারা এটি খুব ভালো বুঝবেন। আমরা যখন কোনো একটি HTML থিম বানাই তার পূর্বে আমাদের একটি PSD ডিজাইন দেয়া হয়। এর মাধ্যমে আমরা বুঝতে পারি HTML দিয়ে আমাদের কি বানাতে হবে। অর্থাৎ এর রং কি হবে, এর মেনু কেমন হবে, এর বাটনগুলো কেমন হবে, এর ইনপুটগুলো দেখতে কেমন হবে এসবই বর্ননা করা থাকে PSD ডিজাইনে। কিন্তু আমরা চাইলেই কিন্তু PSD ডিজাইনকে সরাসরি ব্যবহার করতে পারি না অথবা ব্যবহার করা সম্ভবও না। এর মানে হলো আপনাকে অবশ্যই PSD থেকে HTML এ কনভার্ট করে একে ব্যবহার করতে হবে। আর একে বলা হয় ইমপ্লিমেন্টেশন। এখানে যদি আপনি PSD টিকে প্যারেন্ট ক্লাস ও HTML টিকে চাইল্ড ক্লাসের সাথে তুলনা করেন তাহলেই খেলা খতম। এখানে PSD টিই হলো টোটাল থিমের একটি ইন্টারফেস অর্থাৎ থিমটি কেমন হবে তা এখানেই বর্নানা করা হয়েছে আর প্রোগ্রামিং এর ইন্টারফেসও একই। আপনার একটি প্যারেন ক্লাস থাকবে যেখানে আপনি চাইল্ড কি কাজ করবে। কিন্তু কিভাবে কাজ করবে সেটি বলা থাকবে না। আর ইন্টারফেসে আপনি প্যারেন্ট ক্লাসকে সরাসরি ব্যবহার করতে পারবেন না সেজন্য অবশ্যই একে একবার ইমপ্লিমেন্ট করে নিতে হবে। তাহলে চলুন আমরা Interface এর একটি উদাহরন দেখিঃ

**interface** **Calculation**

{

**public** **function** **sum**($a, $b);

**public** **function** **sub**($a, $b);

}

ইন্টারফেস কে অবশ্যই interface কিওয়ার্ড দিয়ে লিখতে হবে। এটিও একধরনের ক্লাস কিন্তু একে সরাসরি ব্যবহার করা যাবেনা অবশ্য একে ব্যবহারের পূর্বে implement করে নিতে হবে। আর ইন্টারফেসের মেথডগুলোতে কোনো কোড বা ডিটেইল থাকবেনা। শুধু ডিকলারেশন থাকবে। যাইহোক যেহেতু ইন্টারফেস সরাসরি ব্যবহার করা যাবে না, তাই একে ইমপ্লিমেন্ট করে নিতে হবে। চলুন দেখি কিভাবে একে ইমপ্লিমেন্ট করবেনঃ

**class** **Calculator** **implements** **Calculation**

{

**public** **function** **sum**($a, $b)

{

**return** $a + $b;

}

**public** **function** **sub**($a, $b)

{

**return** $a - $b;

}

**public** **function** **div**($a, $b)

{

**return** $a / $b;

}

}

ইমপ্লিমেন্ট করা কিছুটা ক্লাস এক্সটেন্ড করার মতো। শুধু এখানে অবশ্যই আপনাকে ইন্টারফেসে ডিকলার করা মেথডগুলোর ডিটেইল লিখতে হবে অর্থাৎ এর ফাংশনালি দিতে হবে। অবশ্য সাথে আপনি আপনার প্রয়োজনমতোও মেথড বা প্রোপার্টিজ দিয়ে আপনার কাজ চালাতে পারবেন। উপরে দেখুন আমরা Calculator ক্লাসটি Calculation ইন্টারফেস থেকে ইমপ্লিমেন্ট করেছি। আমরা Calculator ক্লাসে ইন্টারফেসের সকল মেথডের ফাংশনালিটিজ দিয়েছি সাথে একটা অতিরিক্ত মেথডও যুক্ত করেছি। এখানে অতিরিক্ত মেথড কিভাবে কি লিখবেন এর কোনো বাধ্যবাধকতা নেই কিন্তু ইন্টারেসে বর্ননা করা সকল মেথডকে ইমপ্লিমেন্টে হুবহু সে রকমের করেই বর্ননা করতে হবে।

Abstract Classes

Abstract class কিছুটা interface এর মতোই এর মূল পার্থক্য হলো এখানে আপনি মেথড এর নাম, ভিজিবিলিটি ও প্যারামিটার ডিকলার করা ছাড়াও চাইলে কিছু মেথডের বর্ননা করে দিতে পারবেন। Abstract class ও আপনি সরাসরি ব্যবহার করতে পারবেন না, একে অবশ্যই নূন্যতম একবার ইনহেরিট করে নিতে হবে। অর্থাৎ abstract class লিখাই হয় ইনহেরিট করে নেয়ার জন্য। চলুন একটি উদাহরন দেখি।

**abstract** **class** **Bird**

{

**abstract** **public** **function** **canFly**();

**abstract** **public** **function** **canSwim**();

**public** **function** **canSound**()

{

**return** **true**;

}

}

খেয়াল করুন পিএইচপিতে abstract class কে abstract কিওয়ার্ড দিয়ে লিখতে হয়। আর যেসকল মেথডগুলোর ব্লুপ্রিন্ট তৈরী করবেন সেগুলোও abstract কিওয়ার্ড দিয়ে লিখতে হবে। এবং আপনি চাইলে এখানে মেথডের বর্ননাও লিখতে পারবেন। আমি এখানে একটি পাখির প্রতিচ্ছবি তৈরী করেছি যেখানে এটি থেকে যে পাখিই তৈরী করা হোকনা কেনো এর অবশ্যই ডানা ও পা থাকতে হবে আর ডিফল্ট এর আমি ভয়েস সিস্টেমও তৈরী করে দিয়েছি। তাহলে চলুন এই Bird ক্লাসটিকে দিয়ে এখন একটি Parakeet(টিয়া) ক্লাস তৈরী করবো যার অবশ্যই ডানা ও পা থাকতে হবেঃ

**class** **Parakeet** **extends** **Bird**

{

**public** **function** **canFly**()

{

**return** **true**;

}

**public** **function** **canSwim**()

{

**return** **false**;

}

**public** **function** **color**()

{

**return** 'green';

}

}

**class** **Duck** **extends** **Bird**

{

**public** **function** **canFly**()

{

**return** **false**;

}

**public** **function** **canSwim**()

{

**return** **true**;

}

**public** **function** **isDomestic**()

{

**return** **true**;

}

}

আমি Bird ক্লাস থেকে Parakeet ও Duck দুটি ক্লাস তৈরী করেছি যার canFly ও canSwim হলো Bird এর পলিমরফিক এনটিটি আর canSound হলো Bird থেকে পাওয়া ডিফল্ট এনটিটি এবং color ও isDomestic কে Parakeet ও Duck ক্লাসের নতুন এককটি এনটিটি হিসেবে যুক্ত করলাম। ফলে এখন Parakeetএ চারটি মেথড কাজ করবে canFly canSwim canSound এবং color ও Duck এর চারটি মেথড canFly canSwimcanSound ও isDomestic. এর ফলে Bird ক্লাস থেকে নতুন দুটি পাখি তৈরী করেছি টিয়া ও হাসকে। যেখানে টিয়া ও হাস উভয়েরই পাখির ডিফল্ট বৈশিষ্ট শব্দ করতে পারার ক্ষমতা রয়েছে। কিন্তু পাখি হলেও এরা এর প্যারেন্টকে ওভাররাইট করেছে ফলে এরা কেউ উড়তে পারে আবার কেউ পারেনা কেউ সাতার পারে আবার কেউ পারে না। আবার নিজেদের কিছু স্বতন্ত্র বৈশিষ্ট্যও রয়েছে।

Polymorphic ক্লাসের এটাই সুবিধা আপনি চাইলে এর মাধ্যেমে ক্লাসের একটি রেস্ট্রিকশন তৈরী করে দিতে পারেন যেটা তার চাইল্ড ক্লাস অবশ্যই ফলো করবে। আধুনিক ফ্রেমওয়ার্কগুলোতে এর প্রচুর ব্যবহার লক্ষ্য করবেন। মর্ডান প্রোগ্রামিং এ এটি অপরিহার্য একটি ফিচার।

এটি ব্যবহারের প্রধান কারন হচ্ছে ভিন্ন ভিন্ন পার্সপেক্টিভে একই ইন্টাফেস ব্যবহার করে আলাদা আউটপুট পাওয়া। এটি এমন একটা ক্ষেত্র যেখানে একাধিক ইমপ্লিমেন্টেশনের একটি নির্দিষ্ট ইন্টারফেস থাকবে অর্থাৎ সবাই একই কনভেনশন ফলো করবে।