```cpp
#pragma once

struct InertiaDiagonal
{
    double Ixx;
    double Iyy;
    double Izz;

    InertiaDiagonal()
        : Ixx(0.0), Iyy(0.0), Izz(0.0)
    {
    }

    // constructor with principal moments (inputs have _in)
    InertiaDiagonal(double Ixx_in, double Iyy_in, double Izz_in)
        : Ixx(Ixx_in), Iyy(Iyy_in), Izz(Izz_in)
    {
    }

    // [Ixx 0 0; 0 Iyy 0; 0 0 Izz]*[wx; wy; wz]
    Vector3 times(const Vector3 &omega_B) const
    {
        return Vector3(
            // Omega_B is [omega_B.x (wx), omega_B.y (wy), omega_B.z (wz)] from input
            Ixx * omega_B.x,
            Iyy * omega_B.y,
            Izz * omega_B.z
            // Output is [Ixx*omega_B.x; Iyy*omega_B.y; Izz*omega_B.z]
        );
    }

    // I inverse times v   for diagonal inertia
    Vector3 inverseTimes(const Vector3 &v_B) const
    {
        return Vector3(
            v_B.x / Ixx,
            v_B.y / Iyy,
            v_B.z / Izz

        );
    }
};
```