

課題 1 パソコンの CPU、マウス、メモリなどのモジュールを管理するクラスをモジュールごとに宣言しなさい。各自 2 つ以上のモジュールを選び、次に示すインタフェース **Module** を実装して宣言すること。下にハードディスクの一例を示す。

```
interface Module
{
    // モジュールカテゴリ CPU, ハードディスク, マウスなどを戻り値として返す
    String getCategory();

    // モジュールの情報 CPUでは製造者、速度などを表示する
    void showInfor();
}

// ハードディスククラス (例)
class HardDrive implements Module // インタフェースの実装
{
    private String name;
    private int capacity;

    HardDrive(String n, int c)
    {
        name=n;
        capacity=c;
    }
    public String getCategory()
    {
        return "HardDrive";
    }
    public void showInfor()
    {
        System.out.println(name+", "+capacity+"GB");
    }
}
```

課題 2 課題 1 で宣言した各クラスのオブジェクトをメインメソッドで生成し、スーパークラスであるインタフェース **Module** 型の配列として作成しなさい。オーバーライドをされたメソッド **getCategory()** と **showInfor()** を用いて配列に格納されている各オブジェクトの情報を出力しなさい。

課題 3 インタフェース **Module** は利用者が作成するサブクラスがもつメソッドの型や引数を指定 (制限) する。そのため、インタフェース **Module** がもつメソッドを用いたコードを予め作成することができる。次はインタフェース **Module** を実装するクラスのオブジェクトを処理するクラス **Manager** である。

```

abstract class Manager
{
    //////////////////////////////////////
    /// 使用法  Manager.listModules(Module[]);
    /// 引 数  インタフェース Module 型の配列変数
    /// 戻 値  無し
    /// 機 能  配列中の各モジュールの一覧出力
    public static void listModules(Module[] m)
    {
        System.out.println("[パーツ一覧/パーツ数:"+m.length+"個]");
        for(int i=0;i<m.length;i++)
        {
            System.out.println("+-----");
            System.out.println("| Parts no."+i);
            System.out.println("Category "+m[i].getCategory());
            m[i].showInfor();
            System.out.println();
        }
    }
    //////////////////////////////////////
    /// 使用法  Manager.selectedModules(Module[], String);
    /// 引 数  インタフェース Module 型の配列変数, カテゴリ名
    /// 戻 値  無し
    /// 機 能  配列中のカテゴリが一致するモジュールの一覧
    public static void selectedModules(Module[] m, String target)
    {
        System.out.println("[選択パーツ一覧/"+target+"]");
        for(int i=0,j=0;i<m.length;i++)
        {
            if(m[i].getCategory()==target)
            {
                System.out.println("+-----");
                System.out.println("| Parts no."+j++);
                System.out.println("Category "+m[i].getCategory());
                m[i].showInfor();
                System.out.println();
            }
        }
    }
    //////////////////////////////////////
    /// 使用法  Manager.countModules(Module[], String);
    /// 引 数  インタフェース Module 型の配列変数, カテゴリ名
    /// 戻 値  カテゴリに一致するモジュールの個数
    /// 機 能  配列中のカテゴリが一致するモジュールの数を出力
    public static int countModules(Module[] m, String target)
    {
        int cnt=0;
        for(int i=0;i<m.length;i++)
            if(m[i].getCategory()==target)cnt++;
        return cnt;
    }
}

```

このクラス宣言を各自のファイルにコピーしなさい。メインメソッド内でこのクラスメソッドを用いて **Module** 型の配列に格納されているオブジェクトの一覧を表示しなさい。