1/21/23 **– Otros tipos de Herencia**

|  |  |
| --- | --- |
| Key Points | Notes |
|  | A partir de ahora las clases que estén siendo heredades las llamaremos familias.  Acabamos de aplicar herencia a la familia Car. Ahora apliquémosla a la familia Payment.  En clases anteriores te mencioné que otro punto de partida que puedes tomar para aplicar herencia es del hecho de que hay clases que lógicamente deberían estar en una familia, como es el caso de Payment.  **Repasemos el diagrama de Payment**  Diagram, schematic  Description automatically generated  Notarás que a nivel de código parece inservible pero cuando estemos en el caso de uso Pagar un Viaje, probablemente en ese momento no sabremos cuál es el método de pago, y necesitemos ingresar un dato lo suficientemente genérico que conceptualmente nos dé la información que necesitamos, en este caso que es un Payment. Este es un tipo de Polimorfismo y uno de los principios SOLID del software que obedece a la Inyección de Dependencias. Lo veremos más adelante a detalle.  Ahora nos faltará crear las clases y aplicar su herencia. |
| Sumario: | |