Herencia, se utiliza para reutilizar clases y ampliarlas a clases más específicas, las estructuras no soportan herencia

P1) Es una propiedad de polygon, una forma de crear una variable, la cual puede ser accedida fácilmente a través de sus getters y setters.

P2) Porque tiene herencia de Polygon, hereda sus características, incluyendo NumberOfSides.

P3) Base y Size, Base contiene la dirección de la clase padre: Polygon, donde esta almacenada NumberOfSides.

P4) No es correcto, no se puede acceder a las variables de la herencia, polygon puede tener la tener la dirección de un square, pero al ser un Polygon no puede acceder a sus miembros (variables).

P5) Si, en polygon está la dirección, pero para poder accederla, se entiende a la variable de referencia como un square, con un cast.

P6) La herencia es en un solo sentido, en una variable de referencia de tipo square no se puede guardar la dirección de un polygon, el hijo no puede tener direcciones de los padres, todos los square son polygon, pero no todos los polygon son square.

P7) Si, se almacena en square la dirección contenida en polygon, entendida como la dirección de un square, se debe hacer esa aclaración aunque este declarada la dirección en polygon como la de un square. El as funciona de la misma forma que el cast.

P8) El programa compila pero saca un error al ejecutar, porque no se puede entender un triangle como un square, no se puede almacenar la dirección de un triangle en un square.

P9)

P10) Si compila.

P11) El “as” permite detectar errores en tiempo de ejecución, si no se puede hacer la conversión devuelve un null.

P12) Si se puede porque number of sides es un miembro de polygon, la clase padre.

P13) 4, con base se pueden alterar los miembros de la clase que vienen de la clase base polygon, con base se llama el constructor de la clase padre.

C1) Con protected se puede acceder a un código siempre que se acceda desde una clase hijo, con public se puede acceder desde fuera de la clase y con private solo se puede acceder dentro de la clase

C2) No se puede acceder a la propiedad porque la clase program no tiene el permiso, y aun siendo esta derivada de polygon no se puede.

P14) Si, porque la referencia a cualquier tipo de objeto puede ser almacenado en un objeto, entonces al utilizar el as, simplemente se entiende que anyOldObject contendrá la dirección de un objeto tipo Square, de forma que se puede acceder directamente a sus miembros.

P15) Partial (tipo) permite dividir la definición de una clase en varios archivos.

P16) C# no soporta herencia múltiple, no se puede.