Challenges and Key Concepts

Part I

>> Challenge 1. Add a class

- *Add a VolleyBall class*
- *Place the class it in a [folder | package] named:* TwoDaysTechIntroExampleClasses
- *Add a constant field to hold the DIAMETER = 21*
- *Use the frmStart class to send the diameter to the console*

OO – KEY CONCEPTS
const (only c#)**:**
- the value is immutable
- It does not change over the life of the program
- It is called "compile-time" value
- only primitive or "built-in" types are allowed to be declared const
- const can't be declared static, by default are static
- needs to get initialized

final (only java):
- can only be assigned once
- (use static final to constants)

>> Challenge 2. Change the code to follow the standard

- *Follow the directions*

TOWA STANDARD – KEY CONCEPTS
class prefix:
- prefix indicating the type of object
- prefixDescriptiveName (camel case)

(StdRef pp 19)
primitive prefix:
- int, long, num, str, bool
- prefixDescriptiveName (camel case)

(StdRef pp 19)
120 max LOC length:
- mandatory

/*TASK xxx*/ ... /*END-TASK*/
- xxx – description of the task

//=============
- to separate types (clasess, enum, …)

col 61 comment:
- //   //comment

>> Challenge 3. Add a calculated static variable and a static constructor

- *Add a private, static numCircumference variable and its get method*
- *Add a private method to calculate the numCircumference variable*
- *Add a static constructor and invoke the method*
- *Send the numCircumference value to the console*

OO – KEY CONCEPTS

static:

- Modifier that can be used with classes, fields, methods, properties, and constructors
- An static class cannot be instantiated, it contains only static members
- A static member belongs to the type itself rather than to a specific object
- You access the members of a static class by using the class name itself

static constructor:

- Initializes static members of the class
- Parameter-less
- Default static constructor initializes static fields to their default value

TOWA STANDARD – KEY CONCEPTS

//--------------

- to separate methods

//- - - - - - - - -

- to separate support methods

_Z suffix

- variable with a property associated

(StdRef pp 19)

function method

- produce a result
- can only include input parameters (_I)
- must not alter any variables
- only one return, the last instruction
- prefixDescriptiveName (camel case)
- should be a prepared for comment method

(StdRef pp 22)

>> Challenge 4. Add an instance variable and a constructor

- *Add an instance private readonly variable named intId and its get method*
- *Add a constructor that receives and integer and set the Id to the object*
- *Create objects and send their Id´s value to the console*

OO – KEY CONCEPTS
constructor:
- Method whose name is the same as the class
- Default constructor instantiates the object and sets member variables to the default values
- It does not include return type
- Its signature includes only the name and its parameter list

readonly (only c#)**:**
- can only be assigned in a *constructor*

format:
- (C#) String.Format,
  {index[:format]}
  format: #,##0,   0.0%
- (java) MessageFormat.format
  {index[,formatType[,subformatPattern]]}
  formatType: number, date, time
  subformatPattern: #,##0,   0.0%

>> Challenge 5. Add a BasketBall class similar to the VolleyBall class

- *Set the const intDIAMETER to 23*
- *Create objects and send their Id´s value to the console*

>> Challenge 6. Add a SoccerBall class similar to the VolleyBall class

- *Set the const intDIAMETER to 21*
- *Create objects and send their Id´s value to the console*

>> Challenge 7. Add an abstract class

- *Name the class BallBallAbstract*
- *Do not forget to use the abstract keyword*
- *Do not forget add the needed comments according to the standard*
- *Make the classes inherit from the Ball class*
- *Declare abstract the numCircumference property in the abstract class and override it in the concrete ones.*
- *Move the intId and the constructor to the abstract class, make the constructor in the concrete classes to call the base constructor*
- *Move the numCalculateCircumference method to the abstract class and make it protected*
- *The numCalculateCircumference should receive the IntDIAMETER as a parameter*
- *The static constructor in the concrete classes should call the method from the abstract class*

OO – KEY CONCEPTS

abstract:

- in a class declaration indicates that is intended to be a base class
- in a method indicates that does not contain implementation
- method marked as abstract must be implemented (overrided) by classes that derive from the abstract class
- An abstract class cannot be instantiated

override:

- An override method provides a new implementation of a member that is inherited from a base class
- The overridden base method must have the same signature
- (c#) override modifier
- (java) @Override

protected:

- member access modifier
- the member is accessible within its class and by derived class instances

TOWA STANDARD – KEY CONCEPTS

_I suffix:

- Add _I to input parameters

>> Challenge 8. Create DUMMY objects from the different classes

- *Add the parameter-less constructor to the concrete classes*
- *Create DUMMY objects and send them to the console*

OO – KEY CONCEPTS

constructor overloading:

- multiple constructors with different signature
- doesn't have a return type

DUMMY:

- It is an object with values without meaning
- Use a parameter-less constructor

>> Challenge 9. Create an object factory for objects

- *Name the object factory method in the abstract class: newball*
- *The newball calls the newballxxx method in the concrete classes*
- *Do not forget to add the abstract declaration of the newballxxx*
- *Do not forget add the override keyword in the newball methods of the concrete classes*

OO – KEY CONCEPTS
object factory:
- mechanism for creating instances of classes
- single place where different objects can be created

>> Challenge 10. Add nullable and enum type variables

- *Add the guest variable, it can be null*
- *Overload the constructor to set the guest value*
- *Add the type and make it an enumeration (VOLLEY, BASKET, SOCCER)*
- *The constructors in the concrete class set the type*

OO – KEY CONCEPTS
nullable:
- int? **-** primitive that can be null

Integer:
- class that wraps a value of a primitive type int in an object
- an object of type Integer contains a single field whose type is int

enum:
- use enum keyword to create enumeration
- a type that consists of a set of named constants
