

Material para a Formación Profesional inicial

A01. Conceptos de POO

Familia profesional	IFC	Informática e comunicacións
Ciclo formativo	CSIFC02 CSIFC03	Desenvolvemento de aplicacións multiplataforma Desenvolvemento de aplicacións web
Grao		Superior
Módulo profesional	MP0485	Programación
Unidade didáctica	UD03	Introdución á POO
Actividade	A01	Conceptos de POO
Autores		Silvia Framiñán Fondevila Marta Rey López
Nome do arquivo		CSIF02_MP0485_V000301_UD03_A01_Conceptos_POO.odt

© 2017 Xunta de Galicia.

Consellería de Cultura, Educación e Ordenación Universitaria.

Este traballo foi realizado durante unha licenza de formación retribuída pola Consellería de Cultura, Educación e Ordenación Universitaria e ten licenza Creative Commons BY-NC-SA (recoñecemento - non comercial - compartir igual). Para ver unha copia desta licenza, visitar a ligazón <http://creativecommons.org/licenses/by-nc-sa/3.0/es/>.

1.	<u>Ficha técnica</u>	5
	<u>Contexto da actividade</u>	5
	<u>Título da actividade</u>	6
	<u>Resultados de aprendizaxe do currículo</u>	6
	<u>Obxectivos didácticos e título e descrición da actividade</u>	6
	<u>Criterios de avaliación</u>	6
	<u>Contidos</u>	6
	<u>Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación (exemplo)</u>	7
2.	<u>A01. Conceptos de POO</u>	8
2.1	<u>Introdución</u>	8
2.2	<u>Actividade</u>	8
2.2.1	<u>Introdución á Programación Orientada a Obxectos</u>	8
2.2.1.1	<u>Características principais da POO</u>	8
2.2.1.2	<u>Terminoloxía de POO</u>	9
	<u>Obxectos</u>	9
	<u>Clases</u>	9
	<u>Encapsulación</u>	10
	<u>Sobrecarga</u>	11
	<u>Polimorfismo</u>	11
	<u>Abstracción</u>	11
2.3	<u>Tarefas</u>	13
2.3.1	<u>Tarefa 1.1. Creación dun diagrama de clases e obxectos</u>	13
	<u>Enunciado</u>	13
	<u>Solución</u>	13
3.	<u>Materiais</u>	14
3.1	<u>Documentos de apoio ou referencia</u>	14
3.2	<u>Recursos didácticos</u>	14
4.	<u>Avaliación</u>	15
4.1	<u>Modelo de proba escrita para CA2.1</u>	15
	<u>PE 1. Creación de diagrama para un caso de POO proposto</u>	15
	<u>PE 2. Cuestións tipo test sobre os conceptos principais da POO</u>	15

1. Ficha técnica

Contexto da actividade

Módulo	Duración	Unidade didáctica.	Sesiões 60'	Actividades	Sesiões 60'
MP0485. Programación.	240	UD01. Identificación dos elementos dun programa informático.	24	A01. Metodoloxía da programación.	5
				A02. Introducción á linguaxe de programación Java.	7
				A03. Elementos básicos dun programa Java.	12
		UD02. Uso de estruturas de control.	20	A01. Estruturas de control selectivas.	8
				A02. Estruturas de control repetitivas.	12
		UD03. Introducción á POO.	28	A01. Conceptos de POO.	8
				A02. Clases e obxectos.	20
		UD04. Conceptos avanzados de POO.	20	A01. Obxectos predefinidos e métodos estáticos.	10
				A02. Constantes e métodos estáticos.	5
				A03. Visibilidade e empaquetaxe.	5
		UD05. Lectura e escritura de información.	34	A01. Fluxos de entrada e saída.	4
				A02. Uso de ficheiros.	15
				A03. Interfaces gráficas de usuario.	15
		UD06. Aplicación das estruturas de almacenamento.	37	A01. Uso de cadeas.	8
				A02. Uso de arrays.	9
				A03. Uso de coleccións e outras estruturas	12
				A04. Manipulación de documentos XML.	8
		UD07. Xerarquías de clases e excepcións.	37	A01. Introducción á herdanza.	10
				A02. Uso avanzado da herdanza.	10
				A03. Interfaces.	10
				A04. Control de código. Excepcións.	7
		UD08. Mantemento da persistencia dos obxectos.	18	A01. Instalación dun SXBDOO e almacenamento básico de obxectos.	8
				A02. Almacenamento e recuperación de obxectos nun SXBDOO e operacións con datos complexos.	10
		UD09. Xestión dos datos almacenados nas bases de datos relacionais.	22	A01. Conexión con sistemas xestores de bases de datos relacionais.	6
				A02. Operacións de lectura nunha base de datos relacional.	7
				A03. Operacións de escritura nunha base de datos relacional e uso de transaccións.	9

NOTA: Esta actividade está vinculada á programación recollida no arquivo CSIFC02_MP0485_V000300_UD03_Introducion_POO.pdf

Título da actividade

Nº	Título	Descrición	Duración
A01	Conceptos de POO.	Nesta actividade analizaranse os fundamentos da POO, poñeranse exemplos do mundo real para explicar os conceptos de clase e obxecto e as súas características e comportamento, asociándoo cos conceptos de atributos e métodos.	8

Resultados de aprendizaxe do currículo

Resultados de aprendizaxe do currículo	Completo
▪ RA2: Escribe e proba programas sinxelos, para o que recoñece e aplica os fundamentos da programación orientada a obxectos.	NON

Obxectivos didácticos e título e descrición da actividade

Obxectivos específicos		Actividade		Descrición básica	Duración
O1.1	Identificar os fundamentos da programación orientada a obxectos.	A01	Conceptos de POO.	Nesta actividade analizaranse os fundamentos da POO, poñeranse exemplos do mundo real para explicar os conceptos de clase e obxecto e as súas características e comportamento, asociándoo cos conceptos de atributos e métodos.	8

Criterios de avaliación

Criterios de avaliación
▪ CA2.1. Identificáronse os fundamentos da programación orientada a obxectos.

Contidos

Contidos
▪ BC2. Uso de obxectos: <ul style="list-style-type: none">– Obxectos: atributos e comportamento.– Características dos obxectos. Tipos de atributos: propiedades.– Estado dun obxecto.– Comportamento dos obxectos: métodos.

Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación (exemplo)

Que e para que	Como			Con que	Como e con que se valora	
Actividade (título e descrición)	Profesorado (en termos de tarefas)	Alumnado (tarefas)	Resultados ou produtos	Recursos	Instrumentos e procedementos de avaliación	
A01. Conceptos de POO. <ul style="list-style-type: none"> Nesta actividade analizaranse os fundamentos da POO, poñeranse exemplos do mundo real para explicar os conceptos de clase e obxecto e as súas características e comportamento, asociándoo cos conceptos de atributos e métodos. 	<ul style="list-style-type: none"> Tp1.1 - Explicación das características principais da POO. 	<ul style="list-style-type: none"> Ta1.1 - Creación dun diagrama de clases e obxectos. 	<ul style="list-style-type: none"> Diagrama de clases e obxectos. 	<ul style="list-style-type: none"> Ordenador persoal con conexión a Internet. Programa para o debuxo de diagramas, por exemplo, Dia. Apuntamentos da profesora. Proxector. 		6
		<ul style="list-style-type: none"> Ta1.2 - Tarefa de avaliación. Exame combinando: <ul style="list-style-type: none"> Creación de diagrama para un caso de POO proposto. Cuestións tipo test sobre os conceptos principais da POO. 	<ul style="list-style-type: none"> Proba escrita. 		<ul style="list-style-type: none"> PE 1. Creación de diagrama para un caso de POO proposto e cuestións tipo test sobre os conceptos principais da POO. 	2

2. A01. Conceptos de POO.

2.1 Introducción

Na actividade que nos ocupa aprenderanse os seguintes conceptos e manexo de destrezas sobre a Programación Orientada a Obxectos:

- Definición e características da Programación Orientada a Obxectos.
- Definición de obxectos e clases.
- Relacións entre clases.
- Explicación dos conceptos de abstracción, sobrecarga, polimorfismo e encapsulación.

2.2 Actividade

2.2.1 Introducción á Programación Orientada a Obxectos

A Programación Orientada a Obxectos (POO) é un paradigma de programación máis próximo ao xeito de expresar os conceptos que empregamos na vida real.

Tradicionalmente empregábase a programación estruturada que consiste en descompoñer o problema obxecto de resolución en subproblemas e estes en máis subproblemas ata chegar a accións moi simples e fáciles de codificar. É dicir, a programación estruturada trata de descompoñer o problema en accións, en verbos. Por exemplo, para calcular o resultado dunha operación primeiro *pídense* os operandos, despois *calcúlase* o resultado e, por último, *amósase* o resultado.

Na POO, pola contra, o problema descomponse en obxectos. En lugar de centrarse nas accións necesarias para resolver o problema, este paradigma céntrase en reproducir o escenario real o máis fielmente posible.

A POO considera a un programa como unha colección de axentes autónomos, chamados obxectos. Mediante a interacción dos obxectos avanza a execución do programa.

2.2.1.1 Características principais da POO

As características principais da Programación Orientada a obxectos son:

- Calquera cousa é un obxecto. Podemos pensar nun obxecto como un tipo especial de variable que almacena datos, pero ao que tamén se lle poden facer peticións para que leve a cabo operacións.
- Un programa é un conxunto de obxectos dicíndose uns a outros o que deben facer mediante o envío de mensaxes. Para facer unha petición a un obxecto, se lle envía a ese obxecto unha mensaxe.

- Cada obxecto ten a súa propia memoria constituída por outros obxectos. Pódese crear un novo obxecto empacotando obxectos existentes.
- Cada obxecto ten o seu tipo, que, en terminoloxía de POO se denominaría clase.
- Todos os obxectos dun tipo particular poden recibir as mesmas mensaxes.

2.2.1.2 Terminoloxía de POO

Antes falabamos de que o mundo real agrupa aos individuos en grupos. Cada tipo de grupo correspóndese a unha **clase** na terminoloxía de POO, e cada individuo, cun **obxecto**. Unha clase é un modelo para un obxecto, no que se define tanto as súas características (mediante os **atributos**) como o seu comportamento ou capacidades (mediante **métodos**).

Obxectos

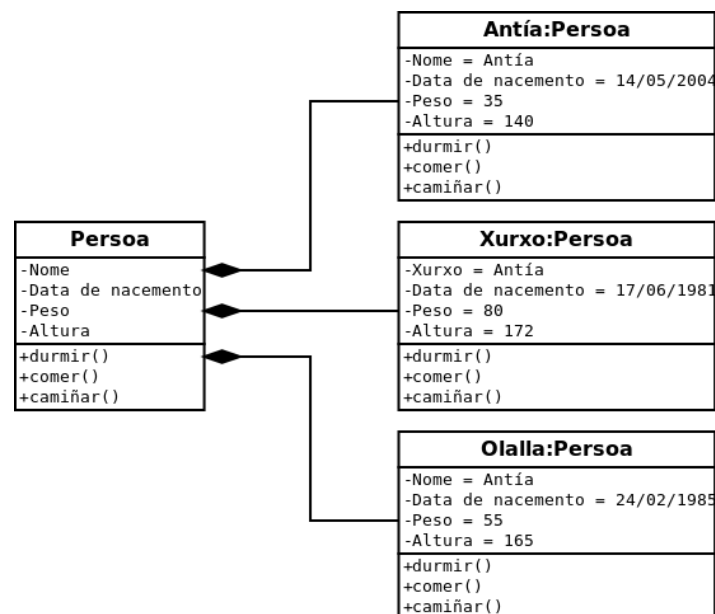
Na vida real, un **obxecto** defínese por unha serie de características, chamadas **propiedades** e que realiza unha serie de operacións (**métodos**).

Cada obxecto é un exemplar dalgunha **clase**, é dicir, unha instancia dunha clase e o seu comportamento queda determinado pola clase á que pertence.

O **estado** dun obxecto comprende todas as súas propiedades e os seus valores actuais.

Os obxectos comunícanse mediante mensaxes. A resposta á mensaxe recibida será a invocación dun método.

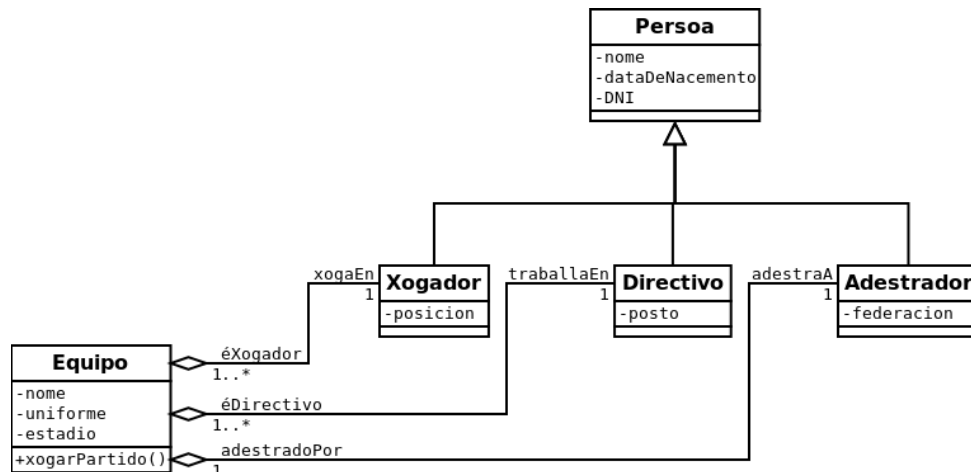
Clases



As clases pódense relacionar entre si de dúas formas:

- **Clientela:** Clases nas cales os seus atributos son outras clases.
- **Herdanza:** Pódense xerar clases como versións especializadas doutras clases.

No seguinte exemplo podemos ver estas relacións. Nun equipo deportivo participan distintos tipos de persoas, entre outras, os xogadores. Temos a clase *Equipo*, que se relaciona coas clases *Xogador*, *Directivo* e *Adestrador* mediante clientela, xa que a clase equipo terá un atributo que almacenará os xogadores que nel xogan, outro cos seus directivos e outro co seu adestrador, que farán referencia a obxectos destas clases. Por outra banda, as clases *Xogador*, *Directivo* e *Adestrador* relaciónanse coa clase *Persoa* mediante herdanza, xa que son tipos de persoas (subclases da clase persoa). Como se pode apreciar na imaxe, ademais dos atributos propios da clase *Persoa*, os obxectos das subclases teñen tamén atributos propios en función do seu subtipo.



Encapsulación

A encapsulación é a forma de protexer os atributos e métodos nas clases, indicando que obxectos poden acceder a eles. Tamén é unha forma de expoñer ao programador unicamente as funcionalidades da clase que lle poden interesar, ocultando a súa complexidade. Existen 3 niveis principais de acceso:

- **Público:** Calquera obxecto de calquera clase pode acceder ao atributo ou método.
- **Protexido:** Só os obxectos desa clase ou de clases descendentes dela poden acceder ao atributo ou método.
- **Privado:** Só o propio obxecto pode acceder ao atributo ou método.

A encapsulación é a maneira que ten unha clase de manter os seus datos seguros, de xeito que non se modifiquen e conteñan datos incoherentes. No exemplo anterior, a clase *Persoa* tiña un atributo DNI, o DNI non pode ser unha cadea de caracteres calquera, se este atributo fose público, quen instancia e emprega o obxecto podería escribir valores erróneos nel, por exemplo “Hola Pepe” ou “1Ae\$39”, ao non selo, a clase non permite que se escriba directamente nel, senón que o fará a través dun método que comprobará que o formato e a letra correspondentes son as adecuadas antes de actualizar o valor.

Sobrecarga

A sobrecarga de métodos é un mecanismo que permite definir nunha clase varios métodos co mesmo nome. Isto resulta útil cando non sempre se lle pasan os mesmos parámetros a un mesmo método e queremos que, segundo o número e tipo de parámetros, se comporte dunha forma ou doutra. Por exemplo, a clase *Empleado* pode ter un método chamado *in-*

crementarSalario que se comporta de dúas formas diferentes: se non se lle pasa ningún parámetro, incrementa o salario do empregado nun 1%, se se lle pasa un parámetro, incrementará o salario na porcentaxe que se lle indique nese parámetro.

Polimorfismo

En programación orientada a obxectos, o polimorfismo é a propiedade pola que é posible enviar mensaxes sintacticamente iguais a obxectos de tipos distintos. O único requisito que deben cumprir os obxectos que se utilizan de maneira polimórfica é saber responder a mensaxe que se lles envía.

Por exemplo, un obxecto da clase *Liña* e un obxecto da clase *Polígono* poderían ter ambos un método *éMaior* ao que se lle pasaría un número enteiro e terían que devolver verdadeiro ou falso. Como se ve, a chamada a ese método das dúas clases distintas é sintacticamente iguais, pero a súa función é diferente. Mentres que o obxecto da clase *Liña* comprobaría se a súa lonxitude é ou non maior que o número pasado como parámetro, o obxecto da clase *Polígono* comprobaría se a súa área é maior que dito número.

Abstracción

A abstracción consiste no illamento dun elemento do seu contexto ou o resto de elementos que o acompañan. En programación, podemos ver cada elemento coma unha caixa negra, da que nos importa “que fai” e non “como o fai”. Poñendo un exemplo da vida real, cando levamos un coche ao mecánico para que o repare, o que nos interesa é que lle entreguemos o coche e nolo devolva reparado, non nos interesa como o fixo. Así, en programación, a abstracción céntrase en ofrecer uns parámetros (no caso do exemplo, o coche) e obter os resultados esperados (o coche reparado) sen importar o procedemento empregado para obtelos (a técnica empregada na reparación).



Realizar a Tarefa 1.1 “Creación dun diagrama de clases e obxectos”, na que se aplicarán os conceptos expostos para a súa aplicación a un caso real.

2.3 Tarefas

As tarefas propostas para esta actividade son as seguintes:

- Tarefa 1.1. **Creación dun diagrama de clases e obxectos**, na que se aplicarán os conceptos expostos a un caso real.

2.3.1 Tarefa 1.1. Creación dun diagrama de clases e obxectos

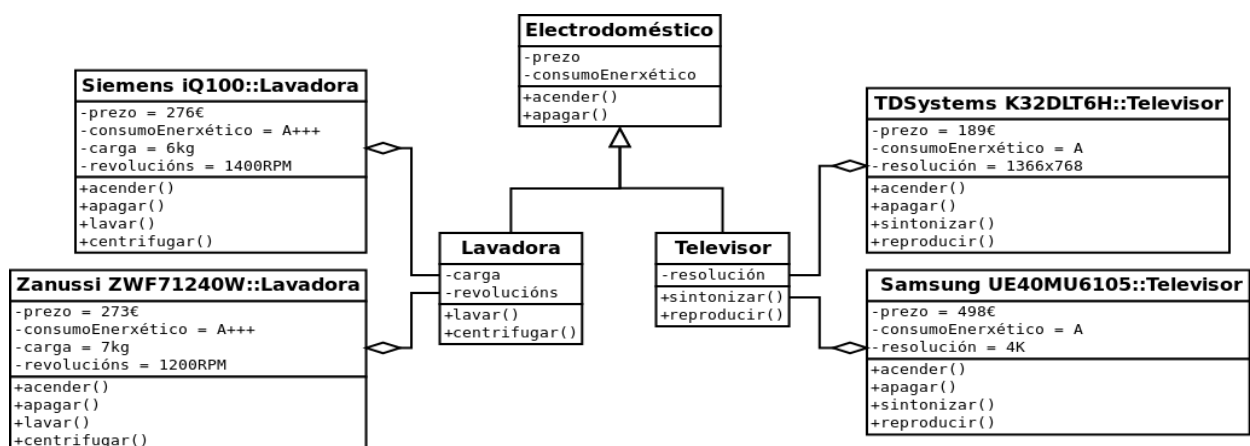
Nesta tarefa aplicaranse os conceptos expostos a un caso real, deseñando as clases e obxectos precisos e plasmándoos nun diagrama.

Enunciado

O propietario dunha tenda de electrodomésticos quere empregar o paradigma da Programación Orientada a Obxectos para modelar a súa tenda. Crea un diagrama no que plasmes mediante clases e obxectos os elementos da tenda (podes poñer só uns poucos exemplos). Non esquezas poñer exemplos das súas propiedades e métodos.

Solución

Para resolvelo, o alumno poderá xerar un diagrama de clases e obxectos coma os expostos no módulo de Contornos de Desenvolvemento. No caso de non ter cursado ese módulo, poderá xerar un diagrama non normalizado ou, simplemente, un listado de clases e obxectos coas súas propiedades e métodos, xa que o relevante desta tarefa non é que se cree o diagrama senón que se identifiquen as clases, obxectos, as súas relacións, propiedades e métodos.



3. Materiais

3.1 Documentos de apoio ou referencia

- Especificación da linguaxe Java: <http://docs.oracle.com/javase/specs/>
- Java SE 8 API Documentation: <https://docs.oracle.com/javase/8/docs/api/index.html>
- The Java Tutorials: <https://docs.oracle.com/javase/tutorial/>
- Conceptos Básicos de Programación Orientada a Objetos. <http://codejavu.blogspot.com.es/2013/05/conceptos-de-programacion-orientada.html>
- V. RESÚA EIRAS. POOJava. <http://iespazodamerce.es/wiki/index.php?title=POOJava>
- P.A. SZNAJDLEDER. Java a fondo. Alfaomega. 2ª edición.
- J. BOBADILLA SANCHO. Java a través de ejemplos. Ra-Ma. Ed. 2003.
- Wikipedia. <https://www.wikipedia.org>

3.2 Recursos didácticos

- Ordenador persoal con conexión a Internet.
- Programa para o debuxo de diagramas, por exemplo, Dia.
- Apuntamentos da profesora.
- Proxector.

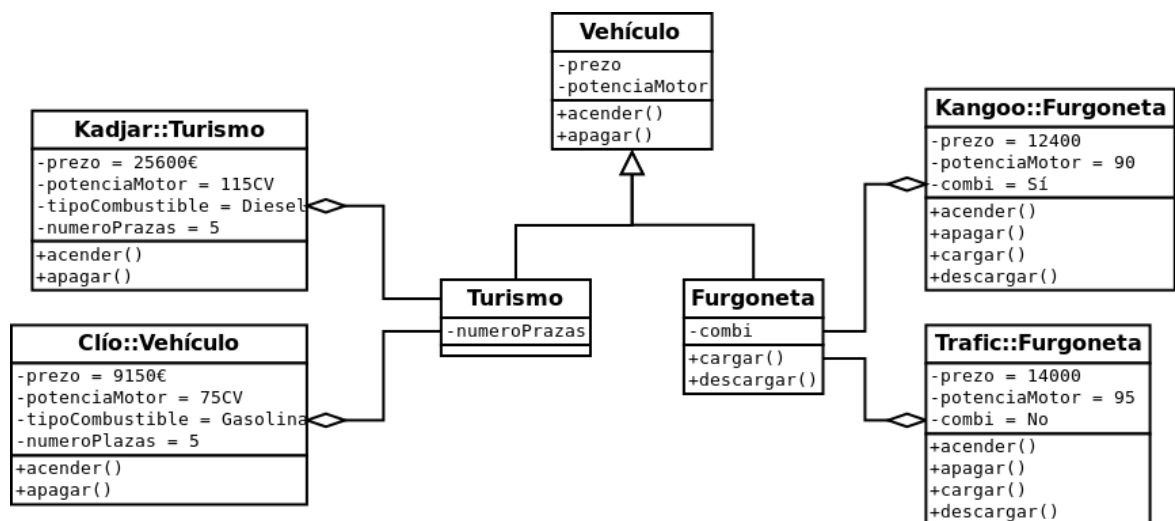
4. Avaliación

Criterios de avaliación seleccionados para esta actividade	Evidencia de aprendizaxe	Instrumento de avaliación	Peso na cualificación da UD
<ul style="list-style-type: none">CA2.1. Identificáronse os fundamentos da programación orientada a obxectos.	<ul style="list-style-type: none">Proba escrita	<ul style="list-style-type: none">PE 1. Creación de diagrama para un caso de POO proposto e cuestións tipo test sobre os conceptos principais da POO.	20%

4.1 Modelo de proba escrita para CA2.1

PE 1. Creación de diagrama para un caso de POO proposto.

- Queremos empregar o paradigma da Programación Orientada a Obxectos para modelar un concesionario de vehículos, no que se venden turismos e furgonetas. Crea un diagrama no que plasmes mediante clases e obxectos os elementos do concesionario (podes poñer só uns poucos exemplos). Non esquezas poñer exemplos das súas propiedades e métodos.



PE 2. Cuestións tipo test sobre os conceptos principais da POO.

- As propiedades dun obxecto

- ☐ Son as operacións que pode realizar.
- ☒ Son as características de dito obxecto.
- ☐ As dúas primeiras opcións son correctas.
- ☐ Ningunha das opcións anteriores é correcta.

2. O estado dun obxecto

- ☒ Comprende todas as súas propiedades e os seus valores actuais.
- ☐ Comprende todas as operacións que realiza.
- ☐ Comprende todas as clases ás que pertence.
- ☐ Ningunha das opcións anteriores é correcta.

3. As relacións entre clases poden ser

- ☐ De irmandade e parentesco.
- ☐ De filiación e irmandade.
- ☒ De clientela e herdanza.
- ☐ Ningunha das opcións anteriores é correcta.

4. A abstracción en POO

- ☐ Modela cada obxecto coma unha caixa negra.
- ☐ Céntrase en “que fai”.
- ☐ Non presta importancia a “como se fai”.
- ☒ Todas as anteriores son correctas.

5. A sobrecarga de métodos é

- ☒ O uso de métodos do mesmo nome con distintos parámetros.
- ☐ O emprego de varias clases ás que pertence un obxecto.
- ☐ A herdanza múltiple.
- ☐ Ningunha das opcións anteriores é correcta.

6. O polimorfismo defínese como

- ☐ O emprego dunha clase con distintas formas.
- ☐ O uso de métodos do mesmo nome con distintos parámetros.
- ☐ O emprego de distintas aparencias para un obxecto.
- ☒ Ningunha das opcións anteriores é correcta.