**PROVA AVALUATIVA**

NOM DE L’ACTIVITAT/PROJECTE/PROVA:

Activitat: Modelatge i Implementació d’un Sistema de Gestió de Projectes

Entrega: **Diagrama de Classes i fitxer de python amb l’implementació comentat.**

Data Entrega: 4/12/2024

RESULTATS D’APRENENTATGE I RA:

RA1. Dissenya diagrames de classes i identifica les relacions d'herència i composició.

1.1. Identifica els conceptes bàsics de l’herència i la composició en programació orientada a objectes.

1.2. Crea un diagrama de classes que inclogui almenys 6 classes, mostrant les relacions d'herència i composició.

1.3. Representa correctament atributs, mètodes i visibilitat en cada classe del diagrama.

1.4. Justifica el disseny seleccionat, explicant com s'aplica a la resolució del problema.

RA2. Implementa un sistema utilitzant el paradigma orientat a objectes (OOP).

2.1. Defineix correctament les classes base i subclasses, respectant els principis d’herència.

2.2. Crea constructors (\_\_init\_\_) que inicialitzin correctament els atributs.

2.3. Utilitza mètodes i propietats per accedir i modificar els atributs d’objectes.

2.4. Aplica el concepte de composició, associant objectes d’una classe a altres objectes.

RA3. Gestiona les funcionalitats del sistema mitjançant mètodes adequats.

3.1. Desenvolupa mètodes que implementin funcionalitats clau, com afegir, eliminar i mostrar informació d'objectes.

3.2. Garanteix que els mètodes de cada classe són autònoms i utilitzables en diferents contextos.

3.3. Defineix mètodes especials (\_\_str\_\_ o similars) per representar objectes com a cadenes de text.

3.4. Verifica que els mètodes implementats es corresponen amb els requeriments especificats.

ENUNCIAT:

Crea un diagrama de classes per a un sistema que gestioni projectes de desenvolupament de programari. El sistema ha de permetre diferenciar entre:

* Projecte: Classe base amb atributs comuns (nom, durada, llenguatge principal).
* ProjecteIntern: Subclasse amb atributs com responsable i departament.
* ProjecteExtern: Subclasse amb atributs com client i pressupost.
* Equip: Representa l'equip que treballa en un projecte, amb atributs com el nom de l'equip i els membres.
* Membre: Classe que representa un membre de l'equip, amb atributs com el nom, el rol i l’experiència.
* Tasques: Classe que gestiona tasques assignades al projecte, incloent el títol, l’estat i el responsable.

Posteriorment, implementa el model en Python i comprova el seu funcionament amb un programa principal.

**Passos a Seguir**

1. Dissenya el diagrama UML que representi les relacions d'herència i els atributs entre les classes.

https://www.lucidchart.com/pages/examples/uml\_diagram\_tool

1. Implementa el disseny en Python.
2. Comprova el funcionament del programa executant-lo i verificant que imprimeix correctament la informació.

**Programa Principal**

if \_\_name\_\_ == "\_\_main\_\_":

# Crear un projecte intern

projecte\_intern = ProjecteIntern(

nom="Aplicació CRM Interna",

duracio=12,

llenguatge="Python",

responsable="Joan Rovira",

departament="IT"

)

# Crear un projecte extern

projecte\_extern = ProjecteExtern(

nom="Plataforma E-learning",

duracio=18,

llenguatge="Java",

client="Educorp",

pressupost=300

)

# Crear un equip i membres

equip = Equip("Equip Desenvolupament")

membre1 = Membre("Anna", "Desenvolupadora", 3)

membre2 = Membre("Marc", "Tester", 2)

equip.afegir\_membre(membre1)

equip.afegir\_membre(membre2)

# Afegir tasques al projecte intern

tasca1 = Tasca("Definir requeriments", "pendent", membre1)

tasca2 = Tasca("Provar funcionalitats", "pendent", membre2)

projecte\_intern.afegir\_tasca(tasca1)

projecte\_intern.afegir\_tasca(tasca2)

# Mostrar informació del projecte intern

print("Informació del projecte intern:")

print(projecte\_intern.mostrar\_informacio())

print("\nTasques del projecte intern:")

print(projecte\_intern.mostrar\_tasques())

# Mostrar informació de l'equip

print("\nInformació de l'equip:")

print(equip.mostrar\_informacio())

print("\nMembres de l'equip:")

print(equip.mostrar\_membres())

# Mostrar informació del projecte extern

print("\nInformació del projecte extern:")

print(projecte\_extern.mostrar\_informacio())

**Sortida Esperada**

Informació del projecte intern:

Projecte: Aplicació CRM Interna, Duració: 12 mesos, Llenguatge: Python, Responsable: Joan Rovira, Departament: IT

Tasques del projecte intern:

Tasca: Definir requeriments, Estat: pendent, Responsable: Anna

Tasca: Provar funcionalitats, Estat: pendent, Responsable: Marc

Informació de l'equip:

Equip: Equip Desenvolupament, Membres: 2

Membres de l'equip:

Membre: Anna, Rol: Desenvolupadora, Experiència: 3 anys

Membre: Marc, Rol: Tester, Experiència: 2 anys

Informació del projecte extern:

Projecte: Plataforma E-learning, Duració: 18 mesos, Llenguatge: Java, Client: Educorp, Pressupost: 300K€

CRITERIS D’AVALUACIÓ:

| **Criteris d'Avaluació** | **Excel·lent (10)** | **Bé (7-9)** | **Millorable (5-6)** | **Insuficient (<5)** |
| --- | --- | --- | --- | --- |
| **Disseny del diagrama UML** | Diagrama complet, precís i ben estructurat. | Diagrama complet però amb petits errors. | Diagrama incomplet o amb molts errors. | Diagrama absent o incorrecte. |
| **Implementació de les classes** | Estructura i herència correctes, constructors ben definits i mètodes funcionals. | Estructura correcta amb petits errors en constructors o mètodes. | Errors en l'herència o mètodes incomplets. | Implementació errònia o absent. |
| **Funcionalitat del programa principal** | Demana dades i mostra resultats correctament. | Demana dades i mostra resultats amb petits errors. | Funciona parcialment. | No funciona o absent. |
| **Claredat i estil del codi** | Codi net, comentat i estructurat. | Codi comprensible però amb millores d'estil. | Codi desordenat o poc comprensible. | Codi confús o absent. |