|  |  |  |
| --- | --- | --- |
| A logo with black text  Description automatically generated | **Academic year: 2023 - 2024**  **Written Partial Exam** | |
| **Module:** Object-Oriented Design and Java Programming  **Grades:** 3B1  **Duration:** 1 hour  **Number of pages:** 6 | | **Documents Allowed:** No  **Internet Allowed:** No  **Date:** 16-03-2024  **Time:** 11:00 a.m. |
|
|

**Please answer on the exam sheet.**

**Exercise I: MCQ (4pts)**

**Only one answer is correct.**

1. Given this class, what is the correct statement to implement a parameterized constructor that initializes x and y?



1. public Mobile (int x, int y) {

x=x;

this.y=y;

}

1. public Mobile (int x, int y) {

super();

this.y=y;

}

1. public Mobile (){

super;

}

1. public Mobile (int x, int y ) {

this.x=x;

this.y=y;

}

1. What is the output of this program ?

****

1. true true
2. false false
3. true false
4. false true
5. Based on this code:



Which of the following definitions in C2 is a correct overriding of the method “meth” of class C1 and not an overloading?

1. public int meth (String s, int n) {return super.meth(s, n \* 2) + 10;}
2. public int meth (int n, String s) {return super.meth(s, n \* 2) + 10;}
3. public int meth (String s, int n, int n2) {return super.meth(s, n\*2) + n2;}
4. public float meth (String s, int n) {return super.meth(s, n \* 2) + 10;}
5. Based on this code, the variable "name" can be accessed directly from:

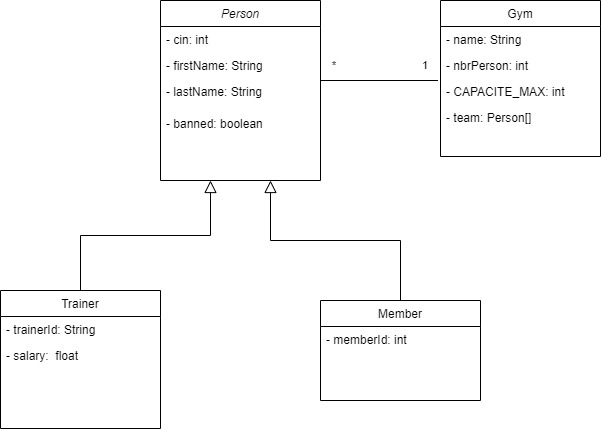


1. any class
2. only the Student class
3. Any class in the package “entities”
4. any class that inherits from the Student class

**Exercise II: Problem (16 pts)**

**Copy the code onto the exam sheet.**

As part of an academic project, you are tasked with developing a program in Java that implements a gym, relying on the class diagram provided and following the instructions given (TODO) by your supervisor.

**Nb: Respect the Java Naming Rules (Conventions).**

**PS-1: The use of the collections (List, Set and Map) is strictly prohibited.**

**PS-2: Note that all the attributes of all classes have the "private" access level.**

1. TODO 1 (1pt): The "Person" class should allow inheritance only for the "Trainer" and "Member" classes.
2. TODO 2 (1pt): Implement the constructor that takes all attributes as a parameter.
3. TODO 3 (1pt): Implement the **"toString()"** method (Don't forget annotations and naming rules)
4. TODO 4 (1pt): Declare the **method "public void showLabel( )"** knowing that it must be redefined in all child classes and not in the Person class.

|  |
| --- |
| **Person Class** |
| **TODO 1**  public................................................ class Person ................................................................ {  private int cin;  private String firstName, lastName;  private boolean banned;  /\* Getters and Setters are already implemented \*/  **TODO 2**  .....................................................................................................................................................  **TODO 3**  .....................................................................................................................................................  **TODO 4**  .....................................................................................................................................................  } |

1. What is the signature of the accessor(getter) of the "banned" attribute, knowing that it is of type boolean? (0.5pt)

...........................................................................................................................

1. TODO 5 (0.5 pt): The "Trainer" class can be extended.
2. TODO 6 (1pt): Implement the constructor that takes all attributes as a parameter.
3. TODO 7 (2pts): Implement salary and trainerId getters and settersknowing that **salary must be greater than or equal to 0** and **trainerId is not empty**.
4. TODO 8 (0.5pt): Redefine the "**public void showLabel( )"** method, knowing that for a trainer, it displays the message "I'm a professional trainer" (Don't forget to add the appropriate annotation)

|  |
| --- |
| **Trainer Class** |
| **TODO 5**  public.......................................... class Trainer ..................................................................... {  private String trainerId;  private float salary;  **TODO 6**  .....................................................................................................................................................  **TODO 7**  .....................................................................................................................................................  **TODO 8**  .....................................................................................................................................................  } |

1. TODO 9 (0.5pt): The "Member" class cannot be extended.
2. TODO 10 (1.5pts): Redefine the **"equals(Object obj)"**  method according to the member's cin and ID.
3. TODO 11 (0.5pt): Redefine the **"public void showLabel( )"** method, knowing that for a member, it displays the message with the memberId. (Don't forget to add the appropriate annotation)

***Example: "I'm a member and my id is 123A45"***

1. TODO 12 (1.5pt): Implement the **"toString( )"** method that reuses the same method of the **Person** class

|  |
| --- |
| **Member Class** |
| **TODO 9**  public.................................... class Member ............................................................ {  private String memberId;  /\* constructors, getters and setters are already implemented \*/  **TODO 10**  .....................................................................................................................................................  **TODO 11**  .....................................................................................................................................................  **TODO 12**  ..................................................................................................................................................... } |

1. TODO 13 (0.5pt): Complete the declaration of CAPACITE\_MAX which is set to 30 and then add the necessary instruction to declare a set that includes coaches and members and which is called "team".
2. TODO 14 (1pt): Create the parameterized constructor that takes as a parameter that initializes the number of members to 0 and creates the group of "team" people.
3. TODO 16 (2pt): The **"public void getStatistics( )"**  method is used to display the number of members and the number of coaches in the team. Banned people will not be considered.

***Example: This gym has 20 members and 4 trainers***

|  |
| --- |
| **Gym Class** |
| public class Gym {  private String name;  private int nbrPersons;  /\* Getters, setters are already implemented \*/  **TODO 13**  private String name;  private .............................. int CAPACITE\_MAX ............................................................;  ...................................................................................................................................................  **TODO 14**  ......................................................................................................................................................  **TODO 15**  public void getStatistics( ) {  ...................................................................................................................................................... } |