## Gym

Create the following project structure



### Class Customer

**Upon initialization** each customer will receive the following **parameters**: **name: str**, **address: str**, **email: str**. Each customer should also have a personal **id** (**autoincremented**, staring form 1). To do the incrementation you should create a **class attribute** **id** equal to **1**,which will keep the value of the **id for the upcoming customer**. For example, if there are **no customers**, the class **id** should be equal to **1**, when there **is one customer** – the class **id** should be equal to **2**.

Create a method called **get\_next\_id** which returns the **id** that will be given to the **next customer.**

Implement the **\_\_repr\_\_** method so it returns the **info** about the customer in the following format: **"Customer <{id}> {name}; Address: {address}; Email: {email}"**

### Class Equipment

**Upon initialization** the class will receive the following **parameters**: **name: str**. Each equipment should also have an **id** (**autoincremented**,starting from **1**). To do the incrementation you should create a **class attribute** **id** equal to **1**,which will keep the value of the **id for the next equipment's id**.

Create a **method** called **get\_next\_id** which returns the **id** that will be given to the **next equipment.**

Implement the **\_\_repr\_\_** method so it returns the **info** about the equipment in the following format: **"Equipment <{id}> {name}"**

Create a **static method** called **get\_next\_id** which returns the **id** that will be given to the **next equipment.**

### Class ExercisePlan

**Upon initialization** the class will receive the following **parameters**: **trainer\_id: int**, **equipment\_id: int**, **duration: int** (in **minutes**). Each plan should also have an **id** (**autoincremented**, starting from **1**). To do the incrementation you should create a **class attribute** **id** equal to **1**,which will keep the value of the **id for the next plan's id**. Create the following **methods**:

* **from\_hours(trainer\_id:int, equipment\_id:int, hours:int)** – creates **new instance** using the provided information
* **get\_next\_id()** – **static method** that returns the **id** that will be given to the **next plan**
* **\_\_repr\_\_()** – returns the **information** about the plan in the following format: **"Plan <{id}> with duration {duration} minutes"**

### Class Subscription

**Upon initialization** the class will receive the following **parameters**: **date:str**, **customer\_id: int**, **trainer\_id: int**, **exercise\_id: int**. The class should also have an **id** (**autoincremented** starting from **1**). To do the incrementation you should create a **class attribute** **id** equal to **1**,which will keep the value of the **id for the next subscription's id**.

Implement the **\_\_repr\_\_** method so it returns the **info** about the subscription in the following format: **"Subscription <{id}> on {date}"**

Create a **static method** called **get\_next\_id** which returns the **id** that will be given to the **next subscription**

### Class Trainer

**Upon initialization** the class will receive the following **parameters**: **name:str**. The class should also have an **id** (**autoincremented** starting from **1**). To do the incrementation you should create a **class attribute** **id** equal to **1**,which will keep the value of the **id for the next trainer's id**.

Implement the **\_\_repr\_\_** method so it returns the **info** about the trainer in the following format: **"Trainer <{id}> {name}"**

Create a **static method** called **get\_next\_id** which returns the **id** that will be given to the **next trainer.**

### Class Gym

**Upon initialization** the class will **not receive** any **parameters**. However, it should have the following **attributes**: **customers** (**list of customer** objects, **empty** upon initialization), **trainers** (**list of trainer** objects, **empty** upon initialization), **equipment** (**list of equipment** objects, **empty** upon initialization), **plans** (**list of plan** objects, **empty** upon initialization), **subscriptions** (**list of subscription** objects, **empty** upon initialization)

Create the following **methods**:

* **add\_customer(customer: Customer)** – **add the customer** in the customer list, if the customer **is not** already **in it**
* **add\_trainer(trainer: Trainer)** – **add the trainer** to the trainers list, if the trainer **is not** already in it
* **add\_equipment(equipment: Equipment)** – **add the equipment** to the equipment list, if the equipment **is not** already in it
* **add\_plan(plan: ExercisePlan)** – **add the plan** to the plans list, if the plan **is not** already in it
* **add\_subscription(subscription: Subscription)** – **add the subscription** in the subscriptions list, if the subscription **is not** already in it
* **subscription\_info(subscription\_id: int)** – get the **subscription,** the **customer** and **trainer**, the **plan** and the **equipment**. Then **return** their **string representations** each on a **new line.**

### Examples

|  |
| --- |
| **Test Code** |
| from project.customer import Customer  from project.equipment import Equipment  from project.exercise\_plan import ExercisePlan  from project.gym import Gym  from project.subscription import Subscription  from project.trainer import Trainer  customer = Customer("John", "Maple Street", "john.smith@gmail.com")  equipment = Equipment("Treadmill")  trainer = Trainer("Peter")  subscription = Subscription("14.05.2020", 1, 1, 1)  plan = ExercisePlan(1, 1, 20)  gym = Gym()  gym.add\_customer(customer)  gym.add\_equipment(equipment)  gym.add\_trainer(trainer)  gym.add\_plan(plan)  gym.add\_subscription(subscription)  print(Customer.get\_next\_id())  print(gym.subscription\_info(1)) |
| **Output** |
| 2  Subscription <1> on 14.05.2020  Customer <1> John; Address: Maple Street; Email: john.smith@gmail.com  Trainer <1> Peter  Equipment <1> Treadmill  Plan <1> with duration 20 minutes |