Exercises – Part 3: A Ticket System

Upload your code as a .py file, e.g., **`exercises_part_3.py`**, on Moodle.

You can find more on enums here: https://www.geeksforgeeks.org/enum-in-python/

Do not copy and paste code from others or the internet! Find your own solution, there are many different correct ways to complete each exercise.

Task 1: Some Classes and String representation

You are responsible for the design and implementation of a ticket system. Customers create tickets with issues that they face when working with a software system.

A Ticket has

- an **id** (`int`). The id represents the ticket numerically.
- a **description** (`str`), which is a message from the user detailing the problem,
- the **priority** (`IntEnum`) of the ticket. Possible priorities are 1 – Severe, 2 – Significant, 3 – Affecting, 4 – Non-Critical, and 5 – Routine,
- the **status** (`Enum`) of the ticket. The possible status are 'Created', 'In progress', 'Testing', 'Closed'. All new tickets have status 'Created',
- [optionally] a **created** timestamp, when the ticket was created (`datetime.now()`)
- a list of **comments**. Each comment consists of
    - the actual **text** (`str`),
    - the **user** (`str`) that added the comment
    - the timestamp (`datetime`), when the comment was made
- a method `add_comment(text: str, user:str, timestamp: datetime)` to add a new comment to the list of comments.

Two kinds of tickets exist:

- A **Software Ticket** has an **error message** (`str`) and the affected **operating systems** (`Enum`). The possible operating systems are 'Windows', 'macOS', 'Linux', 'Android', or 'iOS'. There might be more than one operating system affected, but at least one is necessary. Ids of software tickets start at 100.000.
- A **Hardware Ticket** has the affected **component** (`str`), its **serial number** (`int`), and, if available, the **error code** (`str`). Ids of hardware tickets start at 200.000.

Design the classes necessary to manage these tickets, write constructors for these classes. Make sure that the classes can represent themselves as strings.

Here are examples of what the string representation of two example tickets may look like:

```
T 200.000: Troubles with printer in main hall
           Created 05/14/23 15:42:41 - Priority 3 (Affecting)
           Component: Printer C.1.08b, s/n: 234-23, error code: None
 Comments: Sarah, 05/14/23 16:08:12: This is the third time this month...

T 100.000: Login problems
           Created on 05/12/23 11:09:01 - Priority 1 (Severe)
           Error message: 'Document not found: 404', Affected OSs: MacOS, Windows
 Comments: Tina, 05/12/23 12:01:34: Android does not have that problem
           Christoph, 05/12/23 14:50:21: Android is developed separately
```

Task 2: Hashable objects

Internally, the Ticket System assigns tickets to various teams. Each team has a list of assigned tickets.

In this task, write two additional classes: **Team** and **Assignments**.

Team has a **name** (`str`) representing the team's name and a list of **members** (`list[str]`), which is a list of all the members of the team.

Assignments (`dict[Team, list[Ticket]]`) is a dictionary, where Team is the key referring to a list of Tickets assigned to a particular team. It has two methods:

1. `add(team: Team, ticket: Ticket)` to add a ticket and
2. a string representation.

If, for example, three tickets for two teams are added and printed:

```
assignments = Assignments()

t1 = Team("1", "Tina", "Philip")
t2 = Team("2", "Theresa", "Mirela")

assignments.add(t1, HardwareTicket("Troubles with printer in main hall",
                Priority.AFFECTING, "Printer C.1.08b", "234-23"))
assignments.add(t2, SoftwareTicket("Login problems", Priority.SEVERE,
                "Document not found: 404", OS.MAC_OS, OS.WINDOWS))
assignments.add(t2, SoftwareTicket("Layout issues", Priority.AFFECTING,
                "ValueError", OS.WINDOWS))

print("All assignments:")
print(assignments)

print("\nAssignments of " + str(t1))
print(assignments.get(t1))
```

The output should look something like this:

```
All assignments:
Team 1 (Tina, Philip):
T 200.000: Troubles with printer in main hall
           Created 05/14/23 15:42:41 - Priority 3 (Affecting)
           Component: Printer C.1.08b, s/n: 234-23, error code: None
 Comments: Sarah, 05/14/23 16:08:12: This is the third time this month...
Team 2 (Theresa, Mirela):
T 100.000: Login problems
           Created on 05/12/23 11:09:01 - Priority 1 (Severe)
           Error message: 'Document not found: 404', Affected OSs: MacOS, Windows
 Comments: Tina, 05/12/23 12:01:34: Android does not have that problem
           Christoph, 05/12/23 14:50:21: Android is developed separately
T 100.001: Layout issues
           Created on 05/14/23 16:25:23, priority 3 (Affecting)
           Error message: 'ValueError', OSs: Windows

Assignments of Team 1 (Tina, Philip)
[T 200.000: Troubles with printer in main hall
           Created 05/14/23 15:42:41 - Priority 3 (Affecting)
           Component: Printer C.1.08b, s/n: 234-23, error code: None
 Comments: Sarah, 05/14/23 16:08:12: This is the third time this month...]
```