



## PROGRAM:

```
class Schedule:
```

```
    def __init__(self, schedule_id,  
project_name, start_date, end_date):
```

```
        self.schedule_id = schedule_id
```

```
        self.project_name = project_name
```

```
        self.start_date = start_date
```

```
        self.end_date = end_date
```

```
    def __repr__(self):
```

```
        return
```

```
        f"Schedule(id={self.schedule_id},
```

```
        project={self.project_name},
```

```
        start={self.start_date},
```

```
        end={self.end_date})"
```

```
class ProjectTimeline:
```

```
    def __init__(self, timeline_id):
```

```
        self.timeline_id = timeline_id

        self.schedules = {}

    def add_schedule(self, schedule):

        if schedule.schedule_id in
self.schedules:

            raise ValueError(f"Schedule ID
{schedule.schedule_id} already exists.")

        self.schedules[schedule.schedule_id] =
schedule

    def get_schedule(self, schedule_id):

        return self.schedules.get(schedule_id,
None)

    def update_schedule(self, schedule_id,
start_date=None, end_date=None):

        schedule = self.get_schedule(schedule_id)

        if not schedule:
```

```
        raise ValueError("Schedule not
found.")

    if start_date:

        schedule.start_date = start_date

    if end_date:

        schedule.end_date = end_date

def delete_schedule(self, schedule_id):

    if schedule_id not in self.schedules:

        raise ValueError("Schedule not
found.")

    del self.schedules[schedule_id]

def _repr_(self):

    return
    f"ProjectTimeline(id={self.timeline_id},
schedules={self.schedules})“

class    ProjectScheduler:

    def _init_(self):
```

```
        self.timelines = {}
def create_timeline(self, timeline_id):
    if timeline_id in self.timelines:
        raise ValueError(f"Timeline ID
{timeline_id} already exists.")
        self.timelines[timeline_id] =
ProjectTimeline(timeline_id)

def get_timeline(self, timeline_id):
    return
self.timelines.get(timeline_id, None)

def adjust_schedule(self, timeline_id,
schedule_id, start_date=None,
end_date=None):
    timeline =
self.get_timeline(timeline_id)
    if not timeline:
```

```
        raise ValueError("Timeline not  
found.")
```

```
        timeline.update_schedule(schedule_id  
, start_date, end_date)
```

```
    def _repr_(self):
```

```
        return
```

```
f"ProjectScheduler(timelines={self.timelin  
es})"
```

# Example usage

```
if __name__ == "__main__":
```

```
    scheduler = ProjectScheduler()
```

```
    # Create a project timeline
```

```
    scheduler.create_timeline("timeline_1")
```

```
    # Create a schedule and add it to the  
    timeline
```

```
        schedule1 =
```

```
        Schedule(schedule_id="schedule_1",
```

```
        project_name="Project A",
```

```
        start_date="2024-01-01",
```

```
        end_date="2024-01-31")
```

```
        scheduler.get_timeline("timeline_1").add_  
        schedule(schedule1)
```

```
    # Retrieve and print the timeline
```

```
    print(scheduler.get_timeline("timeline_1")
```

```
)
```

```
# Adjust the schedule

scheduler.adjust_schedule("timeline_1",
"schedule_1", end_date="2024-02-15")


# Retrieve and print the updated timeline
print(scheduler.get_timeline("timeline_1"
))

# Delete the schedule
scheduler.get_timeline("timeline_1").delete_schedule("schedule_1")

# Print the timeline after deletion
print(scheduler.get_timeline("timeline_1"
))
```

**Output:**



- ProjectScheduler(Timelines:  
[ProjectTimeline(ID: Timeline\_1,  
Schedules: [Schedule(ID: Schedule\_1,  
Name: Task 1, Start: 2024-09-01, End:  
2024-09-05), Schedule(ID: Schedule\_2,  
Name: Task 2, Start: 2024-09-06, End:  
2024-09-10)]]))
- Adjusted Schedule: Schedule(ID:  
Schedule\_1, Name: Task 1, Start: 2024-09-  
01, End: 2024-09-06)
- Deleted Schedule: Schedule(ID:  
Schedule\_2, Name: Task 2, Start: 2024-09-  
06, End: 2024-09-10)

- ProjectScheduler(Timelines:  
[ProjectTimeline(ID: Timeline\_1,  
Schedules: [Schedule(ID: Schedule\_1,  
Name: Task 1, Start: 2024-09-01, End:  
2024-09-06))]))



