LU20.L04 - Eventmanager

event.py

```
"""
Dataclass Event.py
"""
from dataclasses import dataclass
from datetime import datetime

@dataclass
class Event:

    title: str
    start: datetime
    end: datetime

    @property
    def duration(self):
        """
        Gibt die Differenz zwischen end und start als timedelta zurück.
        """
        return self.end - self.start
```

management.py

```
"""
Management.py
"""
from datetime import datetime
from event import Event

def read_date(prompt):
    """
    Liest ein Datum im Format "dd.mm.yyyy hh:mm" ein.
    Bei falschem Format wird eine Fehlermeldung ausgegeben
    und erneut abgefragt.
    """
    while True:
        user_input = input(prompt)
        try:
            dt = datetime.strptime(user_input, '%d.%m.%Y %H:%M')
            return dt
        except ValueError:
            print('Invalid date/time format. Please try again.')

def create_event():
    """
```

```
    Liest Titel, Start- und Enddatum vom Benutzer ein
    und gibt ein Event-Objekt zurück.
    """
    title = input('Enter event title: ')
    start_dt = read_date('Enter start date/time (dd.mm.yyyy hh:mm): ')
    end_dt   = read_date('Enter end date/time (dd.mm.yyyy hh:mm): ')

    event = Event(title, start_dt, end_dt)
    print(f'Event "{title}" has been created.')
    return event


def show_events(events):
    """
    Gibt alle Event-Daten (Titel, Start, End, Duration als timedelta)
im Terminal aus.
    """
    if not events:
        print('No events found.')
        return

    print('List of Events:')
    for e in events:
        print('-----------------------------------------------------')
        print(f'Title: {e.title}')
        print(f'Start: {e.start.strftime("%d.%m.%Y %H:%M")}')
        print(f'End:   {e.end.strftime("%d.%m.%Y %H:%M")}')
        # Duration als Zeitspanne ausgeben, z.B. '5:00:00'
        print(f'Duration: {e.duration}')
    print('-----------------------------------------------------')


def main():
    """
    Main function to manage events.
    Allows the user to create new events, view existing events, or exit
the program.
    """
    events = []
    while True:
        print('1) Create Event')
        print('2) Show Events')
        print('3) Exit')
        choice = input('Enter your choice: ')

        if choice == '1':
            e = create_event()
            events.append(e)
        elif choice == '2':
```

```
            show_events(events)
        elif choice == '3':
            print('Exiting the program. Goodbye!')
            break
        else:
            print('Invalid choice!')


if __name__ == '__main__':
    main()
```

⇒ *GitHub Repo für externe Besucher*

GitHub Repository https://github.com/templates-python/m319-lu20-a04-eventmanager

*Lernende am BZZ **müssen** den Link zum GitHub Classroom Assignment verwenden*

m310-LU20

Kevin Maurizi

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m319/learningunits/lu20/loesungen/eventmanager

Last update: **2025/06/23 07:45**