

# Bug/Incident Event-Tracker

## Klasse: event

**Konstruktor:** reporter, titel, beschreibung, priority  
**Klassenattribute:** timestamp, reporter, titel, beschreibung, priority, isOpen  
 Info: Folgende Klassenattribute als private setzen:  
     timestamp = aktuelle Zei  
     isOpen = True

---

**Methoden:** \_\_str\_\_: timestamp, reporter, titel, priority, Status von isOpen

---

**Getter/Setter:** Nach Bedarf zu implementieren

## Klasse: bug (erbt von event)

**Konstruktor:** reporter, titel, beschreibung, priority, gitHash  
**Klassenattribute:** gitHash im Konstruktor als private angeben  
 Info: Im Kontruktor den Konstruktor der Superklasse aufrufen (event.\_\_init\_\_)

---

**Methoden:** \_\_str\_\_: event.\_\_str\_\_(self), gitHash

---

**Getter/Setter:** gitHash

## Klasse: incident (erbt von event)

**Konstruktor:** reporter, titel, beschreibung, priority, cve  
**Klassenattribute:** cve im Konstruktor als private angeben  
 Info: Im Kontruktor den Konstruktor der Superklasse aufrufen (event.\_\_init\_\_)

---

**Methoden:** \_\_str\_\_: event.\_\_str\_\_(self), cve

---

**Getter/Setter:** cve

## Klasse: person

**Konstruktor:** name, email  
**Klassenattribute:** name, email (im Konstruktor als private)

---

**Methoden:** \_\_str\_\_: name, email

---

**Getter/Setter:** name, email

# Bug/Incident Event-Tracker

- Exemplarische Implementierung eines Bug/Incident-Trackers
  1. Implementieren Sie alle vier Klassen lt. Angabe auf der ersten Seite.
    - Wichtig:
      - „bug“ und „incident“ leiten von „event“ ab.
      - Im Konstruktor von „bug“ und „incident“ ist der Konstruktor der Superklasse korrekt aufzurufen
    - Die jeweiligen `__str__` Funktionen sind mit einer sinnvollen Ausgabe zu implementieren
      - (zB. Datum: 01.01.2019, Titel: Mein Titel, Reporter: <Name>)
      - Die `__str__` Funktionen von „bug“ und „incident“ müssen zusätzlich die `__str__` Funktion der Superklasse aufrufen
      - Alle Instanz Attribute sind lokal (`self.__xxxx`) im Konstruktor zu implementieren Getter und Setter sind wie auf voriger Seite beschrieben zu implementieren
      - Destruktoren sind nicht nötig

# Bug/Incident Event-Tracker

- Implementieren Sie einen Code zum Erfassen von Personen/Bugs/Incidents sowie zum Anzeigen und Schließen von Events
- Führen Sie zwei Listen:
  - personenListe und eventListe
  - Folgende Befehle sollen in der Konsole eingegeben werden können

<b>exit</b>	beendet das Programm
<b>new person</b>	Neues Personenobjekt erstellen und in personenListe speichern (Eingabe über Konsole)
<b>list persons</b>	zeigt Liste der Personen an
<b>new bug</b>	Ein neues Bug-Objekt erstellen und in eventListe speichern (Eingabe über Konsole) <b>Achtung! Responsible = Personenobjekt!</b>
<b>new incident</b>	Ein neues Incident-Objekt erstellen und in eventListe speichern (über Konsole) <b>Achtung! Responsible = Personenobjekt!</b>
<b>list events</b>	zeigt Liste aller Bugs und Incidents an. Wichtig! Geben Sie auch aus, ob es ein Bug oder ein Incident ist (Tipp: <code>if type(x) is bug</code> )
<b>close event</b>	schließt ein Event ( <code>isOpen = False</code> )

# Bug/Incident Event-Tracker

## ■ Hinweise

- die Ausgabe der Personenliste und der Eventliste ist mittels `__str__` Methode auszugeben
  - Bitte sinnvoll ausgeben!
  - Das Programm soll in einer Endlosschleife laufen bis **exit** eingegeben wird
  - Bei Problemen können Sie die Attribute auch sichtbar initialisieren (ohne `__`)

## ■ Tipps zum Ablauf

- zuerst die Klassen mit Attribute und Funktionen erstellen (Grundgerüst)
  - danach Vererbung einbauen
  - User in einer Schleife nach Befehlen abfragen lassen (new bug, exit, ...)
  - ... Programmlogik einbauen

# Ticketsystem für Events

## Möglicher Ablauf I

```
#####  
# Ticketsystem #  
#####
```

```
Befehle: exit | new person | list persons | new bug | new incident | list events | close event
```

```
--> new person
```

```
Name --> Oliver
```

```
E-Mail --> oeigner@fhstp.ac.at
```

```
Person hinzugefügt!
```

```
Befehle: exit | new person | list persons | new bug | new incident | list events | close event
```

```
--> list persons
```

```
Personen
```

```
-----
```

```
0, Name: Oliver Email: oeigner@fhstp.ac.at
```

```
Befehle: exit | new person | list persons | new bug | new incident | list events | close event
```

```
--> new bug
```

```
Wer hat den Bug gemeldet?
```

```
0, Name: Oliver Email: oeigner@fhstp.ac.at
```

```
--> 0
```

```
Titel --> MyBug
```

```
Description --> hehe
```

```
Priorität [0-10] --> 5
```

```
Git-Commit-Hash --> dsfdgfdgfdg
```

```
Bug wurde angelegt!
```

```
Befehle: exit | new person | list persons | new bug | new incident | list events | close event
```

```
--> list events
```

```
Events
```

```
-----
```

```
0, MyBug, 2020-02-28 10:34:35.066309, gemeldet von Name: Oliver Email: oeigner@fhstp.ac.at, ist offen: True, Priorität: 5 | gitHash: dsfdgfdgfdg
```

# Ticketsystem für Events

## Möglicher Ablauf II



```
Befehle: exit | new person | list persons | new bug | new incident | list events | close event
```

```
--> new incident
```

Wer hat den Incident gemeldet?

```
0, Name: Oliver Email: oegner@fhstp.ac.at
```

```
--> 0
```

Titel --> Stromausfall

Description --> weg war es

Priorität [0-10] --> 10

CVE --> 831

Incident wurde angelegt

```
Befehle: exit | new person | list persons | new bug | new incident | list events | close event
```

```
--> list events
```

Events

-----

```
0, MyBug, 2020-02-28 10:34:35.066309, gemeldet von Name: Oliver Email: oegner@fhstp.ac.at, ist offen: True, Priorität: 5 | gitHash: dsfdgfdgfdg
```

```
1, Stromausfall, 2020-02-28 10:35:17.686241, gemeldet von Name: Oliver Email: oegner@fhstp.ac.at, ist offen: True, Priorität: 10 | CVE: 831
```

```
Befehle: exit | new person | list persons | new bug | new incident | list events | close event
```

```
--> close event
```

```
0, MyBug, 2020-02-28 10:34:35.066309, gemeldet von Name: Oliver Email: oegner@fhstp.ac.at, ist offen: True, Priorität: 5 | gitHash: dsfdgfdgfdg
```

```
1, Stromausfall, 2020-02-28 10:35:17.686241, gemeldet von Name: Oliver Email: oegner@fhstp.ac.at, ist offen: True, Priorität: 10 | CVE: 831
```

```
--> 0
```

Event wurde geschlossen!

```
Befehle: exit | new person | list persons | new bug | new incident | list events | close event
```

```
--> exit
```