1. a)

In [ ]:

```
class Podcast:
    'Pod'

    def __init__(self, index, thema, length, moderator, adverts = 0):
        self.index = index
        self.thema = thema
        self.length = length
        self.moderator = moderator
        self.adverts = adverts

    def display(self):
        print('Podcast Folge Nr.: ' + str(self.index)+' mit ModeratorIn: ' + self.mod
        + self.thema + '\n' + 'Laenge der Podcastfolge: ' + str(self.length) + '\n' +
        'Aktuelle verkaufte Werbebloecke: ' + str(self.adverts))

newpodcast = Podcast(23, "Traveling Salesman Problems", 70, "O. Peratio")

newpodcast.display()
```

```
Podcast Folge Nr.: 23 mit ModeratorIn: O. Peratio
Thema der Podcastfolge: Traveling Salesman Problems
Laenge der Podcastfolge: 70
Aktuelle verkaufte Werbebloecke: 0
```

1. b)

In [ ]:

```
class Podcast:
    'Pod'

    def __init__(self, index, thema, length, moderator, adverts = 0):
        self.__index = index
        self.__thema = thema
        self.__length = length
        self.__moderator = moderator
        self.__adverts = adverts

    def display(self):
        print('Podcast Folge Nr.: ' + str(self.__index)+' mit ModeratorIn: ' + self._
        + self.__thema + '\n' + 'Laenge der Podcastfolge: ' + str(self.__length) + '\
        'Aktuelle verkaufte Werbebloecke: ' + str(self.__adverts))

    def setAdverts(self, n):
        if self.__adverts+n <= 3:
            self.__adverts += n
            self.__length += 3*n
        else:
            print(' Achtung: Es dürfen nur noch maximal '+str(3-self.__adverts)+ ' We

    def getAdverts(self):
        return self.__adverts

    def cut(self, m):
        if (self.__length - m < 30):
            print('Achtung: Diese Folge kann nicht weiter gekürzt werden. Aktuell ist
            ' Minuten lang.')
        else:
            self.__length -= m
        return self.__length
```

```
newpodcast = Podcast(23, "Traveling Salesman Problems", 70, "O. Peratio")

newpodcast.display()
```

```
Podcast Folge Nr.: 23 mit ModeratorIn: O. Peratio
Thema der Podcastfolge: Traveling Salesman Problems
Laenge der Podcastfolge: 70
Aktuelle verkaufte Werbebloecke: 0
```

In [ ]:

```
newpodcast = Podcast(23, "Traveling Salesman Problems", 70, "O. Peratio")

newpodcast.display()

newpodcast.setAdverts(2)
newpodcast.setAdverts(2)

newpodcast.cut(45)
newpodcast.cut(10)

newpodcast.display()
```

```
Podcast Folge Nr.: 23 mit ModeratorIn: O. Peratio
Thema der Podcastfolge: Traveling Salesman Problems
Laenge der Podcastfolge: 70
Aktuelle verkaufte Werbebloecke: 0
 Achtung: Es dürfen nur noch maximal 1 Werbeblöcke eingespielt werden.
Achtung: Diese Folge kann nicht weiter gekürzt werden. Aktuell ist sie 31 Minuten lan
g.
Podcast Folge Nr.: 23 mit ModeratorIn: O. Peratio
Thema der Podcastfolge: Traveling Salesman Problems
Laenge der Podcastfolge: 31
Aktuelle verkaufte Werbebloecke: 2
```

1. c)

In [ ]:

```
class SpecialPodcast(Podcast):

    def __init__(self, index, thema, length, moderator, specialguest:str):
        super().__init__(index, thema, length, moderator)
        self.__specialguest  = specialguest

    def setGuest(self, newGuest):
        if type(newGuest) != str:
            pass
        else:
            self.__specialguest = newGuest

    def getGuest(self):
        return self.__specialguest


    def changeGuest(self, newGuest:str):
        self.__specialguest = newGuest

    def display(self): #overwrite
        super().display()
        print(f'Specialguest der Podcastfolge ist: {self.__specialguest}')



sp = SpecialPodcast(1, "Vehicle Routing Problem", 65, "M. Ipler", "Dr. Best")
```

```
sp.changeGuest("Dr. Secondbest")

sp.display()
```

```
Podcast Folge Nr.: 1 mit ModeratorIn: M. Ipler
Thema der Podcastfolge: Vehicle Routing Problem
Laenge der Podcastfolge: 65
Aktuelle verkaufte Werbebloecke: 0
Specialguest der Podcastfolge ist: Dr. Secondbest
```

1. a)

In [ ]:

```
import json

f = open('InputFlowshopSIST.json')

d = json.load(f)

print(d)
print("--------------------------------------------------------------------------------")

dict_jobs = d['Jobs']
print(dict_jobs)
```

```
{'Name': 'InputFlowshopSIST', 'nMachines': 5, 'nJobs': 11, 'Jobs': [{'Id': 1, 'SetupTi
mes': [15, 18, 27, 22, 28], 'ProcessingTimes': [375, 12, 142, 245, 412], 'DueDate': 17
79, 'TardCosts': 200}, {'Id': 2, 'SetupTimes': [45, 44, 27, 49, 37], 'ProcessingTime
s': [632, 452, 758, 278, 398], 'DueDate': 3777, 'TardCosts': 200}, {'Id': 3, 'SetupTim
es': [49, 51, 30, 23, 37], 'ProcessingTimes': [12, 876, 124, 534, 765], 'DueDate': 346
6, 'TardCosts': 200}, {'Id': 4, 'SetupTimes': [17, 32, 47, 30, 45], 'ProcessingTimes':
[460, 542, 523, 120, 499], 'DueDate': 3216, 'TardCosts': 200}, {'Id': 5, 'SetupTimes':
[27, 46, 29, 15, 31], 'ProcessingTimes': [528, 101, 789, 124, 999], 'DueDate': 3812,
'TardCosts': 200}, {'Id': 6, 'SetupTimes': [55, 27, 48, 29, 20], 'ProcessingTimes': [7
96, 245, 632, 375, 123], 'DueDate': 3256, 'TardCosts': 200}, {'Id': 7, 'SetupTimes':
[58, 48, 49, 44, 51], 'ProcessingTimes': [532, 230, 543, 896, 452], 'DueDate': 3980,
'TardCosts': 200}, {'Id': 8, 'SetupTimes': [22, 22, 29, 57, 56], 'ProcessingTimes': [1
4, 124, 214, 543, 785], 'DueDate': 2520, 'TardCosts': 200}, {'Id': 9, 'SetupTimes': [3
7, 49, 41, 56, 37], 'ProcessingTimes': [257, 527, 753, 210, 463], 'DueDate': 3315, 'Ta
rdCosts': 200}, {'Id': 10, 'SetupTimes': [58, 19, 19, 19, 24], 'ProcessingTimes': [89
6, 896, 214, 258, 259], 'DueDate': 3784, 'TardCosts': 200}, {'Id': 11, 'SetupTimes':
[30, 52, 31, 32, 37], 'ProcessingTimes': [532, 302, 501, 765, 988], 'DueDate': 4632,
'TardCosts': 200}]}
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
----
[{'Id': 1, 'SetupTimes': [15, 18, 27, 22, 28], 'ProcessingTimes': [375, 12, 142, 245,
412], 'DueDate': 1779, 'TardCosts': 200}, {'Id': 2, 'SetupTimes': [45, 44, 27, 49, 3
7], 'ProcessingTimes': [632, 452, 758, 278, 398], 'DueDate': 3777, 'TardCosts': 200},
{'Id': 3, 'SetupTimes': [49, 51, 30, 23, 37], 'ProcessingTimes': [12, 876, 124, 534, 7
65], 'DueDate': 3466, 'TardCosts': 200}, {'Id': 4, 'SetupTimes': [17, 32, 47, 30, 45],
'ProcessingTimes': [460, 542, 523, 120, 499], 'DueDate': 3216, 'TardCosts': 200}, {'I
d': 5, 'SetupTimes': [27, 46, 29, 15, 31], 'ProcessingTimes': [528, 101, 789, 124, 99
9], 'DueDate': 3812, 'TardCosts': 200}, {'Id': 6, 'SetupTimes': [55, 27, 48, 29, 20],
'ProcessingTimes': [796, 245, 632, 375, 123], 'DueDate': 3256, 'TardCosts': 200}, {'I
d': 7, 'SetupTimes': [58, 48, 49, 44, 51], 'ProcessingTimes': [532, 230, 543, 896, 45
2], 'DueDate': 3980, 'TardCosts': 200}, {'Id': 8, 'SetupTimes': [22, 22, 29, 57, 56],
'ProcessingTimes': [14, 124, 214, 543, 785], 'DueDate': 2520, 'TardCosts': 200}, {'I
d': 9, 'SetupTimes': [37, 49, 41, 56, 37], 'ProcessingTimes': [257, 527, 753, 210, 46
3], 'DueDate': 3315, 'TardCosts': 200}, {'Id': 10, 'SetupTimes': [58, 19, 19, 19, 24],
'ProcessingTimes': [896, 896, 214, 258, 259], 'DueDate': 3784, 'TardCosts': 200}, {'I
d': 11, 'SetupTimes': [30, 52, 31, 32, 37], 'ProcessingTimes': [532, 302, 501, 765, 98
8], 'DueDate': 4632, 'TardCosts': 200}]
```

In [ ]:

```
class DataMachine:
```

```
    def __init__(self, machineId):
        self.MachineId = machineId

    def __str__(self):
        return 'Machine ' + str(self.MachineId)


class DataJob:

    def __init__(self, idjob, processingTimes, setupTimes, dueDate, tardinessCost):
        self.__JobId = idjob
        self.__ProcessingTime = processingTimes
        self.__DueDate = dueDate
        self.__SetupTime = setupTimes
        self.__TardCost = tardinessCost

    def __str__(self):
        return f'Job: {self.__JobId} with {len(self.__ProcessingTime)} Operations.\n'
```

In [ ]:

```
class Jobs():
    def __init__(self, Id, SetupTimes, ProcessingTimes, DueDate, TardCosts=200):
        self.Id = Id
        self.SetupTimes = SetupTimes
        self.ProcessingTimes = ProcessingTimes
        self.DueDate = DueDate
        self.TardCosts = TardCosts

    def __str__(self):
        print(f'ID: {self.Id}\n')
        print(f'SetupTimes: {self.SetupTimes}\n')
        print(f'ProcessingTimes: {self.ProcessingTimes}\n')
        print(f'DueDate: {self.DueDate}\n')
        print(f'TardCosts: {self.TardCosts}\n')
```