

1. Täydennä seuraava koodi toimivaksi. Kirjoita koodia vain harmaisiin kohtiin. Hyödynnä periytymissuhdetta eli älä kirjoita samaa koodia useaan kertaan. Ohjelman tulee toimia samoin kuin esimerkkitulosteessa.

```
class Person:
    def __init__(self, first_name, last_name):
```

```
        def plot(self):
```

```
# pääohjelma
```

```
p1 = Person("James", "Bond");
s1 = Student("Johnny", "English", 321);
p1.plot();
s1.plot();
```

Ohjelman tuottama tuloste:

```
-- My name is James Bond
-- My name is Johnny English
and my student number is 321
```

2. Määrittele muutamalla lauseella seuraavat olio-ohjelmoinnin käsitteet

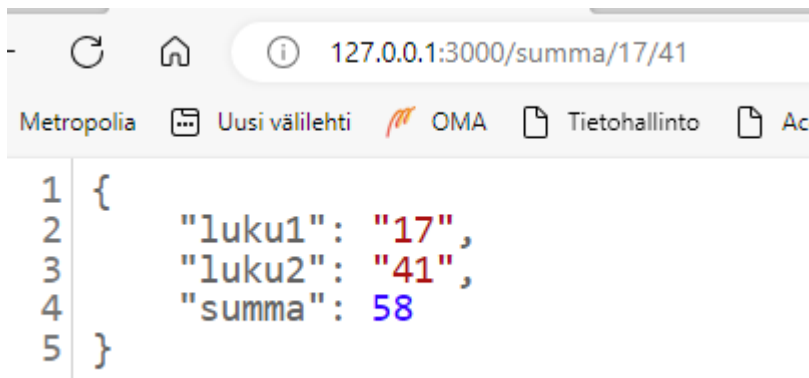
a) luokka / class

b) olio / object

c) periytyminen

d) assosiaatio

3. Alla on kuva selaimesta lähetetystä API-kutsusta ja siihen saadusta vastauksesta. Täydennä ohjelmassa olevat varjostetut kohdat siten, että API tuottaa selaimen kaiun käyttäjän syötteeseen eli tulostaa käyttäjän tekstin perään saman tekstin toisen kerran. Virheisiin ei tarvitse varautua. (max. 8 p.)



```
from flask import Flask, Response
```

```
app = Flask(__name__)  
@app.route('_____  
def kaiku(_____) :
```

```
if __name__ == '__main__':  
    app.run(use_reloader=True, host='_____', port=_____)
```