Exercises

1) Create a Book class with title, author, and year attributes and methods to change the book's title, author, and year.
2) Create a Movie class with title, director, and year attributes and methods to change the movie's title, director, and year.
3) Create a House class with address, bedrooms, and bathrooms attributes and methods to change the house's address, number of bedrooms, and number of bathrooms.
4) Create a BankAccount class with a balance attribute and a method that adds interest to the account balance.
5) Create a Person class with name and age attributes and a method to check if the person is an adult (age >= 18).
6) Create a Rectangle class with height and width attributes and a method to check if the rectangle is a square.
7) Create a Dog class with name, breed, and age attributes and a method to calculate the dog's age in dog years (1 human year = 7 dog years).
8) Create a Cat class with name, color, and age attributes and a method to calculate the cat's age in cat years (1 human year = 4 cat years).
9) Create a Bank class with a list of BankAccount objects and a method to find the account with the lowest balance.
10) Create a Book class with title, author, and year attributes and a method to check if the book was written before a given year.

Exercises and solution

1) Book class:

```
class Book:
    def __init__(self, title, author, year):
        self.title = title
        self.author = author
        self.year = year

    def set_title(self, title):
        self.title = title

    def set_author(self, author):
        self.author = author

    def set_year(self, year):
        self.year = year
```

2) Movie class:

```
class Movie:
    def __init__(self, title, director, year):
        self.title = title
        self.director = director
        self.year = year

    def set_title(self, title):
        self.title = title

    def set_director(self, director):
        self.director = director
```

```
    def set_year(self, year):
        self.year = year
```

3) House class:

```
class House:
    def __init__(self, address, bedrooms, bathrooms):
        self.address = address
        self.bedrooms = bedrooms
        self.bathrooms = bathrooms

    def set_address(self, address):
        self.address = address

    def set_bedrooms(self, bedrooms):
        self.bedrooms = bedrooms

    def set_bathrooms(self, bathrooms):
        self.bathrooms = bathrooms
```

4) BankAccount class:

```
class BankAccount:
    def __init__(self, balance):
        self.balance = balance

    def add_interest(self, rate):
        self.balance += self.balance * rate
```

5) Person class:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def is_adult(self):
```

```
        return self.age >= 18
```

6) Rectangle class:

```
class Rectangle:
    def __init__(self, height, width):
        self.height = height
        self.width = width

    def is_square(self):
        return self.height == self.width
```

7) Dog class:

```
class Dog:
    def __init__(self, name, breed, age):
        self.name = name
        self.breed = breed
        self.age = age

    def dog_years(self):
        return self.age * 7
```

8) Cat class:

```
class Cat:
    def __init__(self, name, color, age):
        self.name = name
        self.color = color
        self.age = age

    def cat_years(self):
        return self.age * 4
```

9) Bank class:

```
class Bank:
    def __init__(self, accounts):
        self.accounts = accounts

    def lowest_balance_account(self):
        if len(self.accounts) == 0:
            return None

        lowest_account = self.accounts[0]

        for account in self.accounts[1:]:
            if account.balance < lowest_account.balance:
                lowest_account = account

        return lowest_account
```

10) Book class:

```
class Book:
    def __init__(self, title, author, year):
        self.title = title
        self.author = author
        self.year = year

    def is_written_before(self, year):
        return self.year < year
```