

195 Zoo

Instructions for the code task:

Product declaration: Data structures, keystrokes and file management, time and date.

You will write a program that guides visitors to a zoo. The visitor must be able to enter the date and time interval for the visit, and the program must print out which animals are expected to be awake, and when these animals are fed (if the feeding occurs during the visit). Here's what it might look like:

```
What date do you want to visit Neptune Zoo? 21 June
```

```
The zoo is open between 06-23.
```

```
What time do you want to come? 13-16
```

```
During your visit you can see:
```

```
Bear
```

```
Sea lion *** fed at 14 ***
```

```
Seal *** is fed at 14 ***
```

```
Wolf
```

```
Moose
```

Information about the animals' waking hours must be stored on file, for example in the following format

```
Format: name / hibernation / time off / feed time
```

```
Bear / winter / 9-20 / 12
```

```
Latmask / summer / 12-14 / 13
```

```
Night owl / - / 21-05 / 21
```

```
Sea Lion / - / 6-18 / 14
```

```
Seal / - / 6-18 / 14
```

```
Wolf / - / 6-20 / 12
```

```
Moose / - / 7-19 / 10
```

You do NOT need to introduce error checking for all inputs. It is ok for the program to crash when entered incorrectly.

Specification task:

Before writing the program, you must write and submit a specification. The purpose of the specification is for you to think through the task properly. The specification shall include the following:

Algorithm. An overview of how the problem can be solved algorithmically. This is a description for a human, not a computer.

Data structure. What data structure or structures do you need? Most often, data needs to be stored in some way; in memory (object instance of a class) or in files (serialized with CSV, JSON or other suitable format). What does your class (es) look like (attributes and methods)? Which of Python's built-in structures do you need?

Features. You can divide your algorithm into functions that, with the help of the arguments they receive, solve the different parts of the algorithm.

It should provide an overview of the project, so quality is more important than quantity: You can see an example of what the specification should look like in the attached file "Spec_example".

Assessment:

- Informative prints / easy-to-understand GUI
- Easy entry
- Comments
- Descriptive name
- Consistent language / typography
- Have at least one class
- Division into functions / methods
- Parameters / return value
- No code repetition
- Flexibility (easy to make changes)
- Meets the requirements for the task
- Important: do not need to implement error handling!

THE PROGRAM SHOULD BE FOR A BEGINNER, EASY CODED AND EASY TO READ!