

Lab 7.5.1 Exceptions: using in program

Objectives

Familiarize the student with:

- using exceptions in real programs;
- writing his/her own exception classes.

Scenario

Write a simple container class that holds values representing the Tower of Hanoi. The Tower of Hanoi is a simple puzzle where the user/player has three rods (towers) and a number of disks of different sizes which can be moved between the rods. Only one disk at a time can be taken from the top of one of the towers and moved to another tower where there are no disks or where all the disks (on the target tower) are bigger than the one being moved. The number of disks is not specified, but to simplify your task, write the simplest version with three (3) disks only. Add some simple input code to test that when the user inputs 12, it is a command to move a disk from the 1st to the 2nd tower. Add three exception classes, all to inform the user that the input is not correct:

- the first when one or both towers are not in the 1–3 range;
- the second to inform the user that the disk moved is of an incorrect size (bigger than the one in the target rod);
- the third to inform the user that there are no disks at the source rod.

Throw these exceptions when the appropriate situation occurs. Write a piece of sample code to solve this puzzle (the minimum version is seven moves – as shown in the example input). Just one note: when the user inputs 11, 22, or 33, it's just an empty (but correct) move. Write a function to print the content of all the towers in a simple format: tower_number disk_sizes. The starting position is like this:

- tower_1: 3 2 1;
- tower_2: empty;
- tower_3: empty.

Print the content of all towers after the last move.

Example input

```
13
12
32
13
21
23
13
```

Example output

```
tower_1: empty
tower_2: empty
tower_3: 3 2 1
```