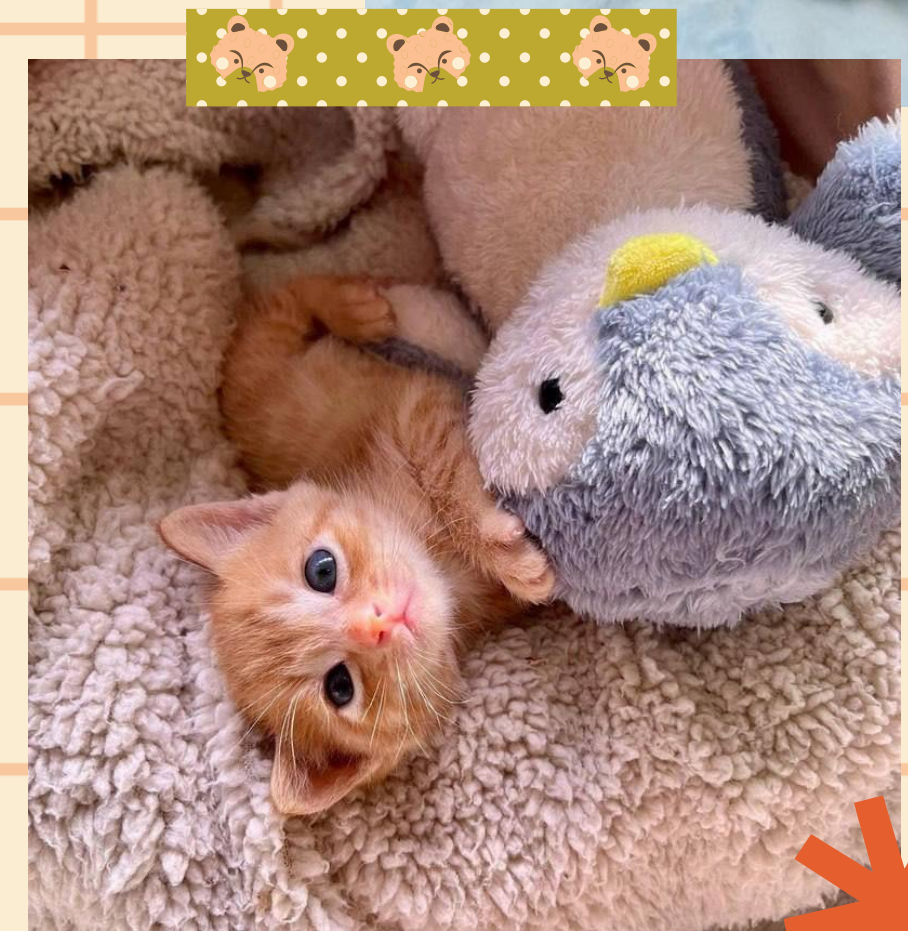
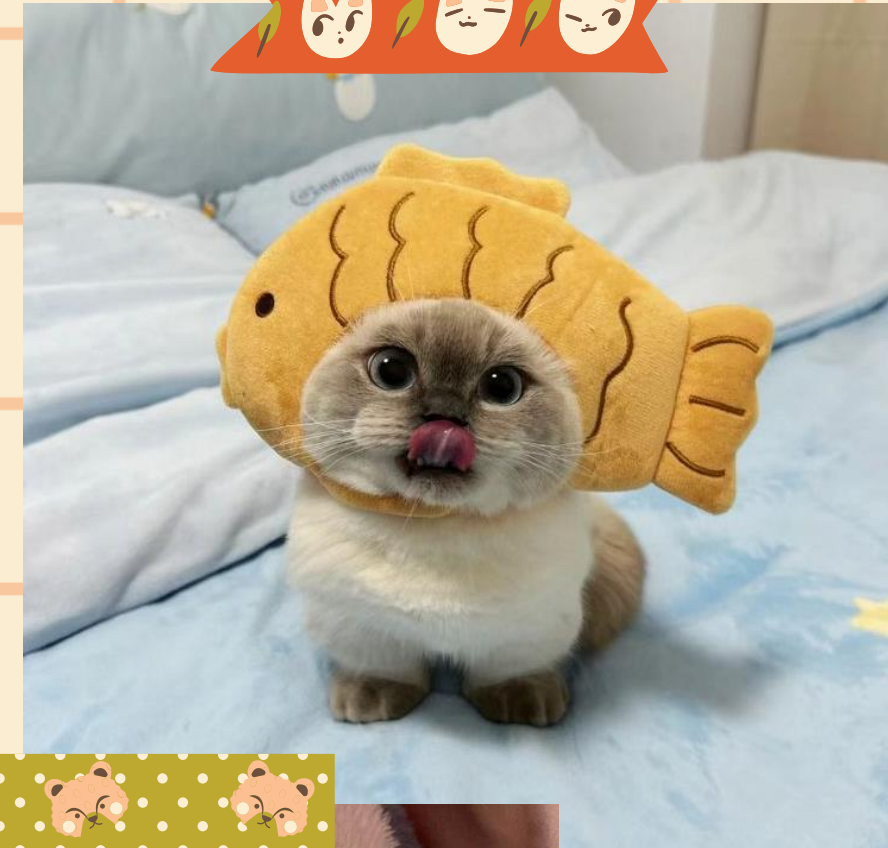


WELCOME TO OUR PRESENTATION

# GOLDEN CALCULATOR

Small Golden Cock (SGC) Team





# Our team



Maksim Kalutski



"Testing, GUI, Math,  
Documentation,  
Presentation"

Volodymyr Kaznacheiev



"Math, GUI, Installer"

Murad Mikogaziev



"Math, Profiling, Makefile"

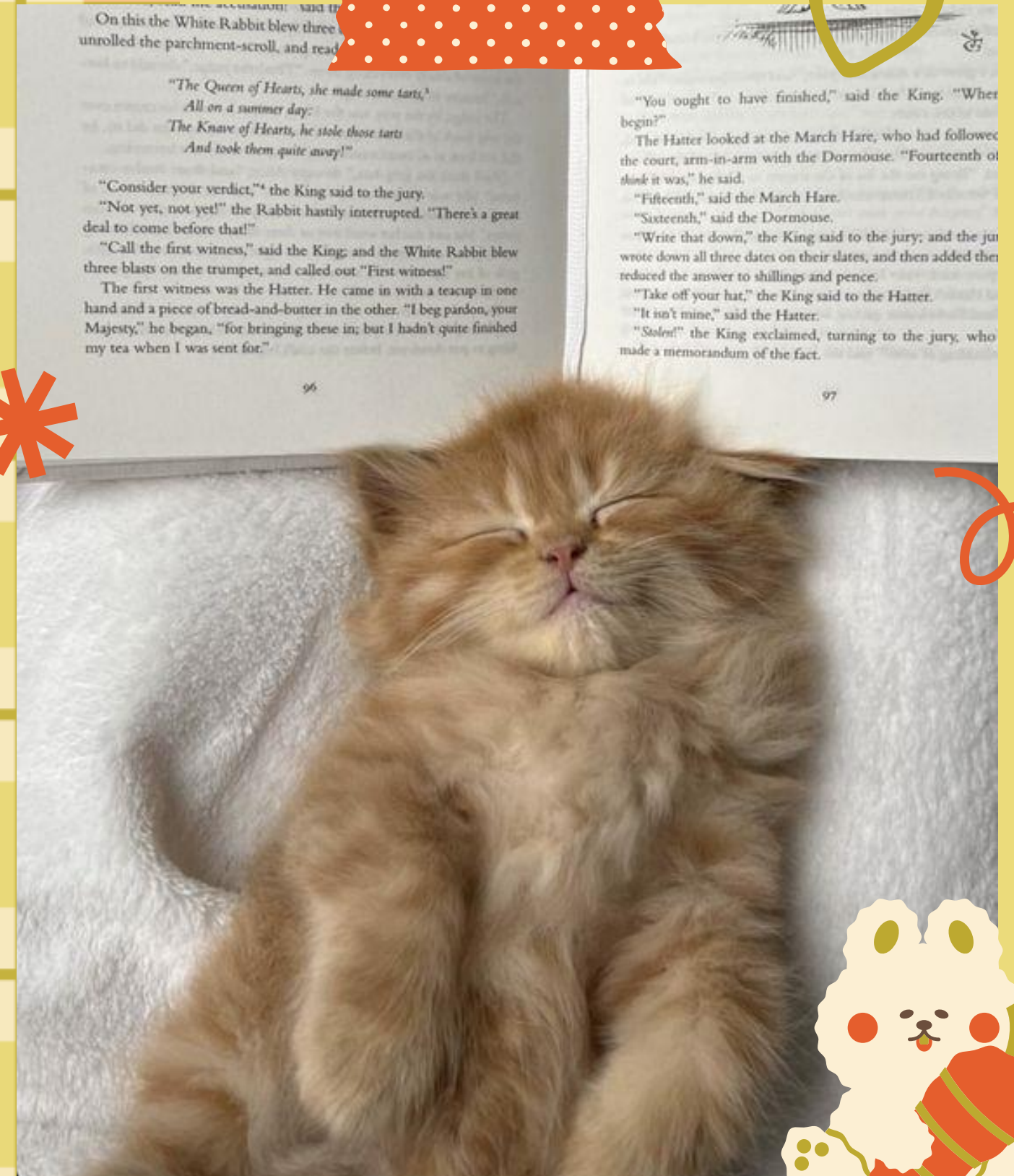
Mikhail Pushkarev



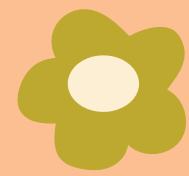
"No task assigned"



# GUI



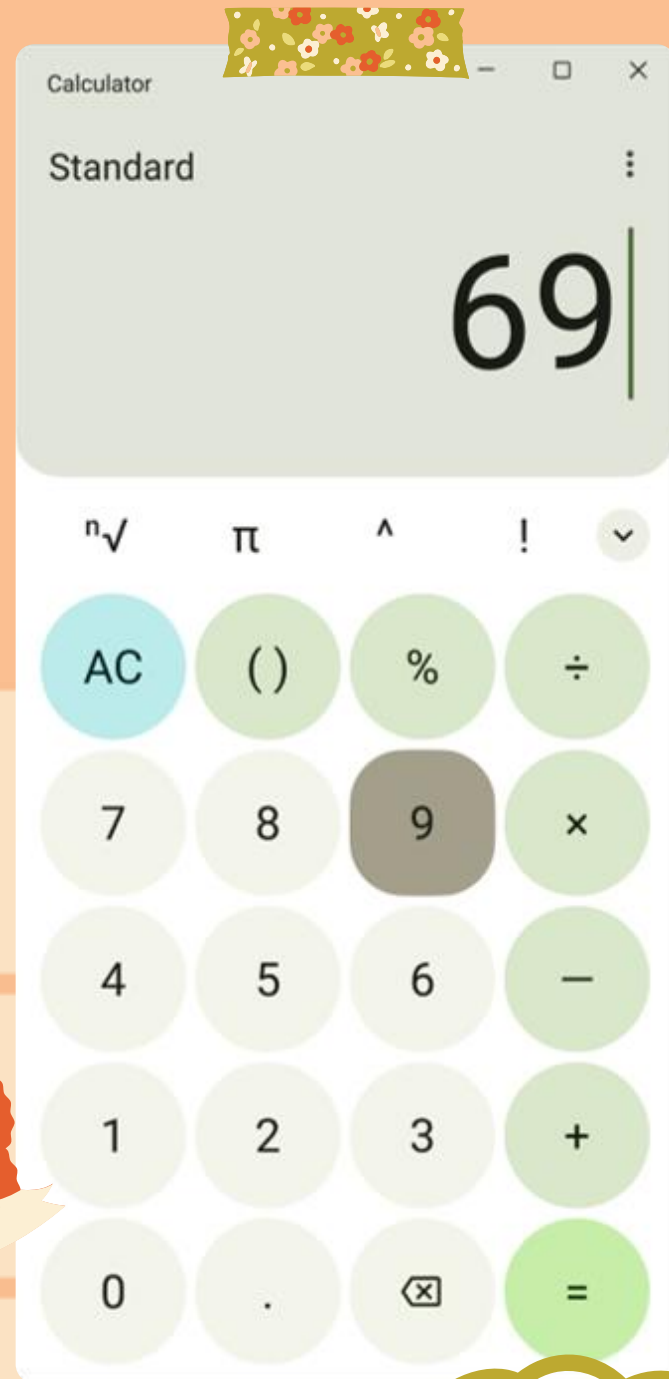




# Design



## Mockup



## Final





# Tkinter





# GUI Module

- tkinter window
- display
- current expression
- buttons grid



# Help



Help



## Calculator Usage Instructions:

Use the numeric keys to enter values.  
Press 'AC' to clear the display.  
Press 'DEL' to delete the last entry.  
To use the square root, press '√' followed by the number (e.g., '√9').  
To calculate an nth root, type the degree followed by '√' and the number (e.g., '3√8').  
If no number precedes '√', it defaults to square root.

OK



6/25



# Testing







Pytest

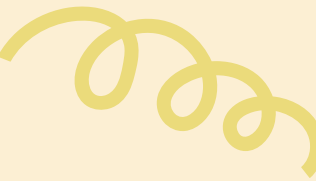


pytest





# Test Case Coverage



```
✓ Test Results 0ms ✓ Tests passed: 134 of 134 tests - 0ms

calc_test.py::test_complex_expressions[0.33333 + 0.66667-1.0] PASSED [ 91%]
calc_test.py::test_complex_expressions[1.000001 - 0.000001-1.0] PASSED [ 91%]
calc_test.py::test_complex_expressions[2.000001 * 1.999999-4.000002] PASSED [ 92%]
calc_test.py::test_complex_expressions[(3! + 4) * 2-20] PASSED [ 93%]
calc_test.py::test_complex_expressions[2 ^ 3! + 4-68] PASSED [ 94%]
calc_test.py::test_complex_expressions[1 / 0-Error: Division by zero] PASSED [ 94%]
calc_test.py::test_complex_expressions[2 ^ 0-1] PASSED [ 95%]
calc_test.py::test_complex_expressions[(-2) ^ 2-4] PASSED [ 96%]
calc_test.py::test_complex_expressions[(-2) ^ 3--8] PASSED [ 97%]
calc_test.py::test_complex_expressions[8 ^ (1/3)-2] PASSED [ 97%]
calc_test.py::test_complex_expressions[9 ^ (1/2)-3] PASSED [ 98%]
calc_test.py::test_complex_expressions[27 ^ (-1/3)-0.3333333333333333] PASSED [ 99%]
calc_test.py::test_complex_expressions[16 ^ 0.5-4] PASSED [100%]

===== 134 passed in 0.23s =====

Process finished with exit code 0
```







# Documentation

10/25





# DOXYGEN

## IVS Project 2 - Calculator 1.0

Calculator project for IVS course at FIT

[Main Page](#)[Namespaces ▾](#)[Classes ▾](#)[Files ▾](#)

### Namespace List

Here is a list of all namespaces with brief descriptions:

[detail level 1 2 3]

▾ **N** src

▾ **N** app

**C** CalculatorApp

**N** config

**N** math\_logic

**N** profiling

Generated by **doxygen** 1.9.1



# Math Logic





# Infix to Postfix

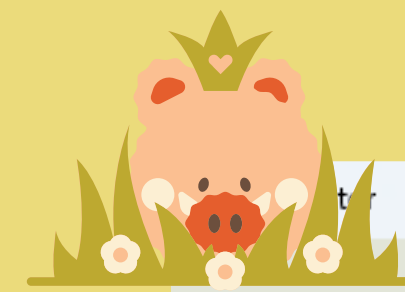
$1 + 2 * 3 \rightarrow 1 2 3 * +$





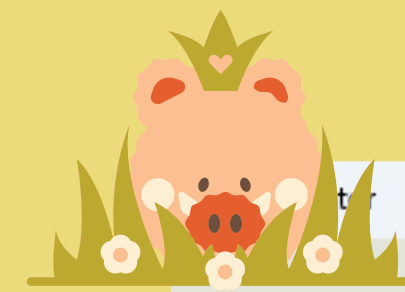
# Arithmetic Functions

- addition
- subtraction
- multiplication
- division
- percentage



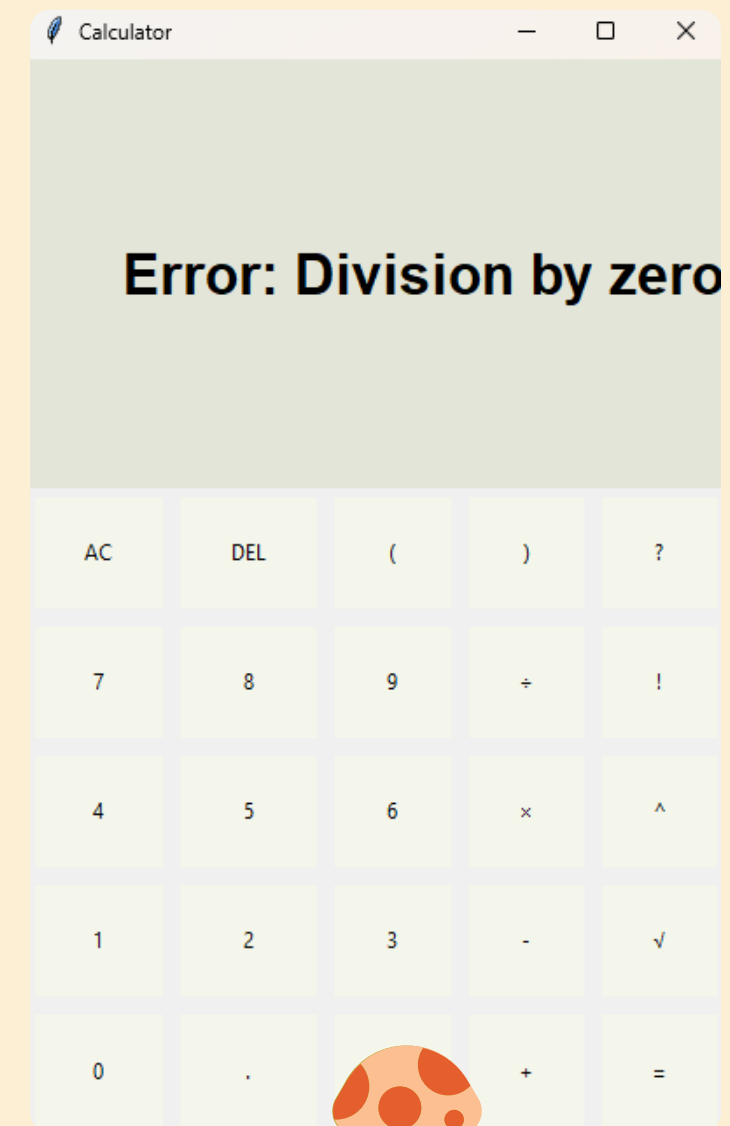
# Advanced Functions

- factorial
- power
- root

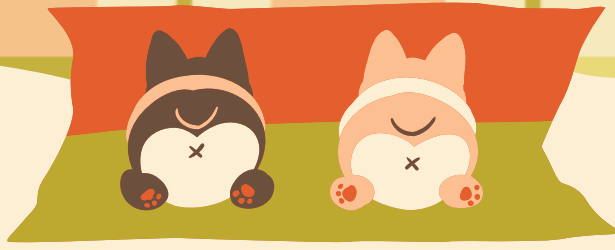




# Error Handling



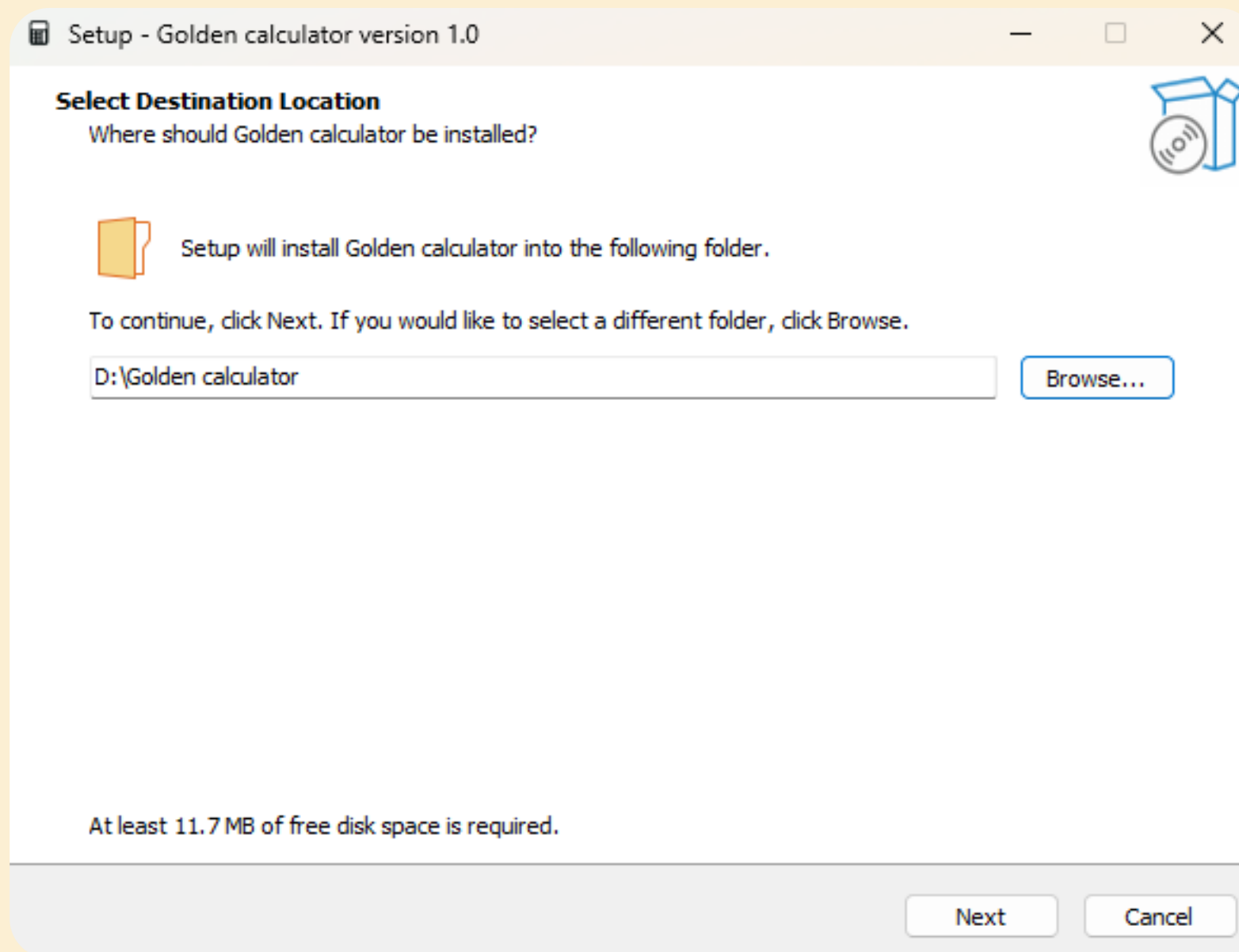




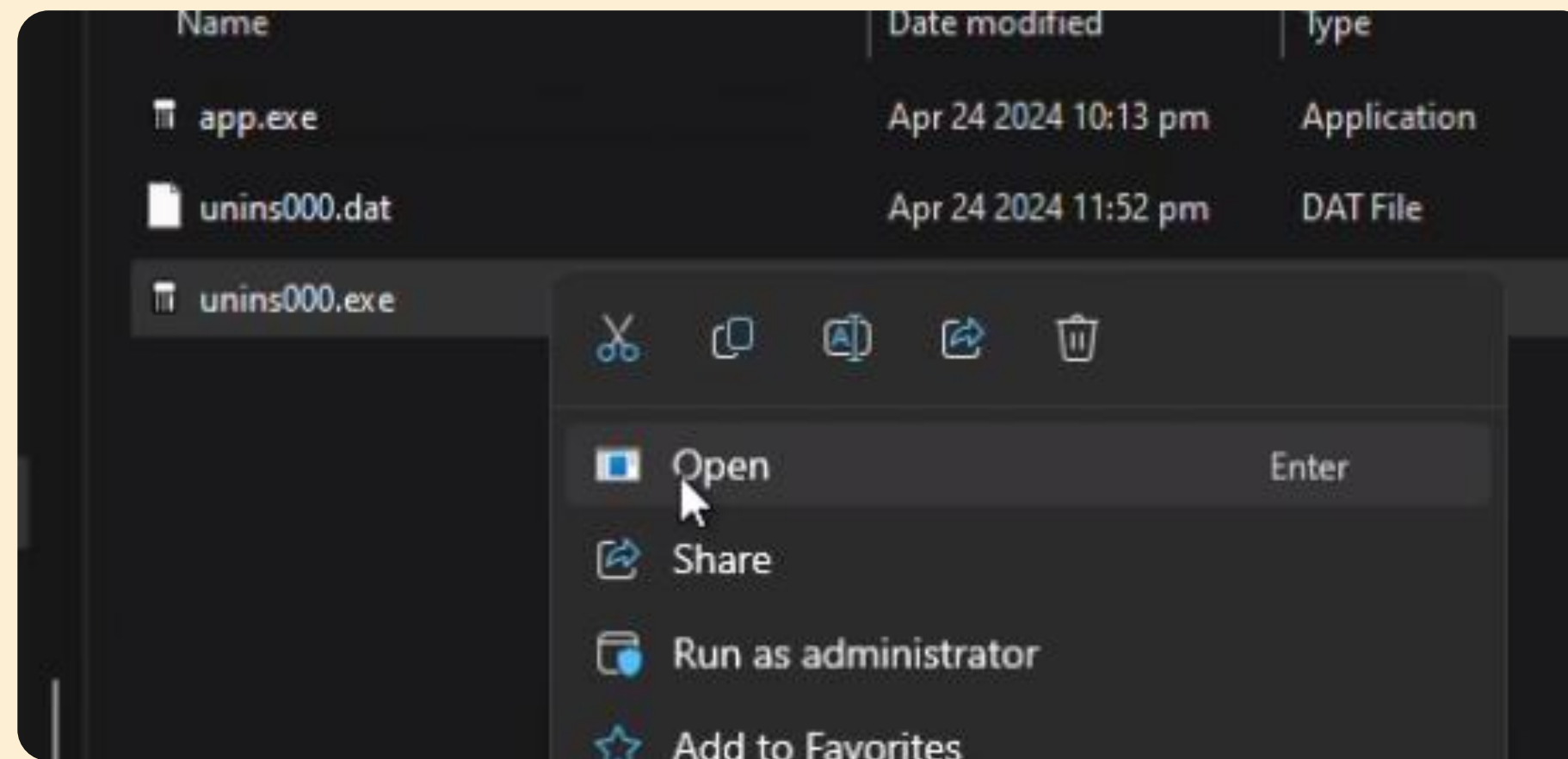
# Installation and uninstallation



# INSTALLING



# UNINSTALLING





# Profiling

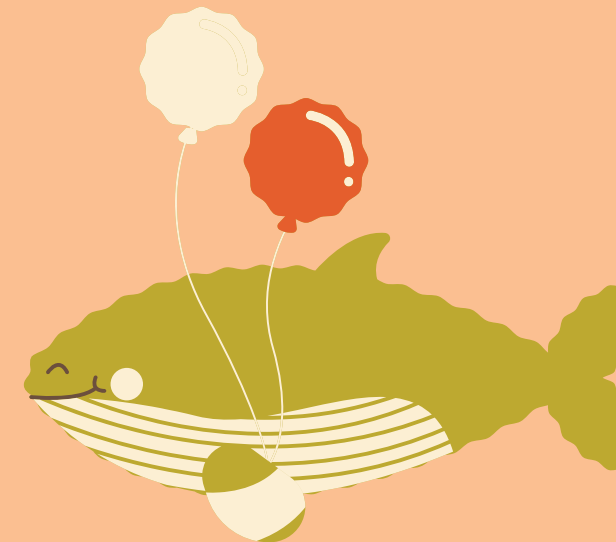


# PROFILING

```
Mean: 15.0
    24 function calls in 0.001 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1   0.000   0.000   0.001   0.001 <string>:1(<module>)
      2   0.000   0.000   0.000   0.000 math_logic.py:144(plus)
      4   0.000   0.000   0.000   0.000 math_logic.py:156(minus)
      2   0.000   0.000   0.000   0.000 math_logic.py:182(div)
      3   0.000   0.000   0.000   0.000 math_logic.py:195(power)
      1   0.000   0.000   0.000   0.000 math_logic.py:223(root)
      1   0.001   0.001   0.001   0.001 profiling.py:15(calculate_sample_std_deviation)
      4   0.000   0.000   0.000   0.000 profiling.py:33(<genexpr>)
      1   0.000   0.000   0.001   0.001 {built-in method builtins.exec}
      1   0.000   0.000   0.000   0.000 {built-in method builtins.len}
      1   0.000   0.000   0.000   0.000 {built-in method builtins.print}
      2   0.000   0.000   0.000   0.000 {built-in method builtins.sum}
      1   0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
```





# Standard deviation

$$s = \sqrt{\frac{1}{N-1} \left( \sum_{i=1}^N x_i^2 - N\bar{x}^2 \right)}$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$







# Makefile





- all
- pack
- clean
- test
- doc
- run
- profile





Thanks for  
listening folks!



Ben Fatto!

