Лабораторная работа 4.4

Тема: Работа с исключениями в языке Python

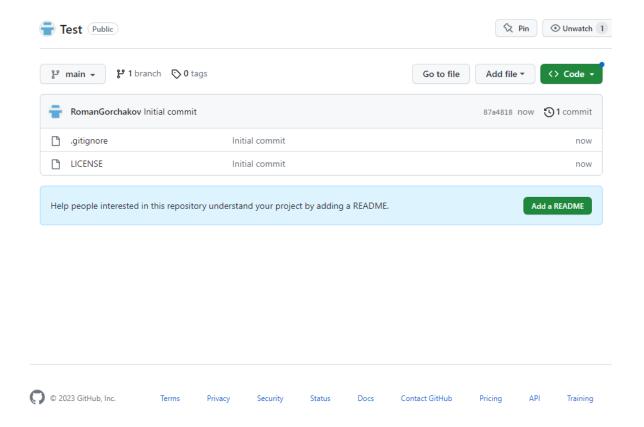
Цель работы: приобретение навыков по работе с исключениями при написании программ с помощью языка программирования Python версии 3.х.

Ссылка на GitHub: https://github.com/RomanGorchakov/Py3_5

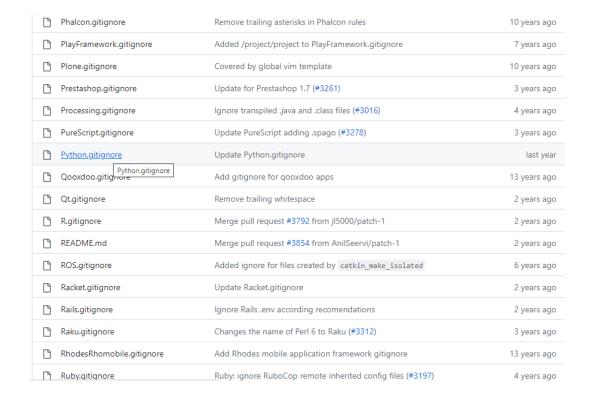
Порядок выполнения работы

1. Создаём аккаунт в GitHub. Затем создаём новый общедоступный репозиторий, в котором будет использована лицензия МІТ и язык программирования Python.

Owner *	Repository name *
RomanGorchakov 🕶	
	/ Test
	Test is available.
Great repository names are sho	ort and memorable. Need inspiration? How about automatic-octo-chainsaw ?
Description (optional)	
Public	
Anyone on the internet	can see this repository. You choose who can commit.
O A Private	
— You choose who can see	e and commit to this repository.
Initialize this repository with:	
Add a README file	
This is where you can write a lor	ng description for your project. Learn more about READMEs.
Add .gitignore	
.gitignore template: Python ▼	
	a list of translation from a consideration of the
Choose which files not to track from	a list of templates. Learn more about ignoring files.
Choose a license	
License: MIT License ▼	
	1. 10.1. 20
A license tells others what they can a	and can't do with your code. Learn more about licenses.

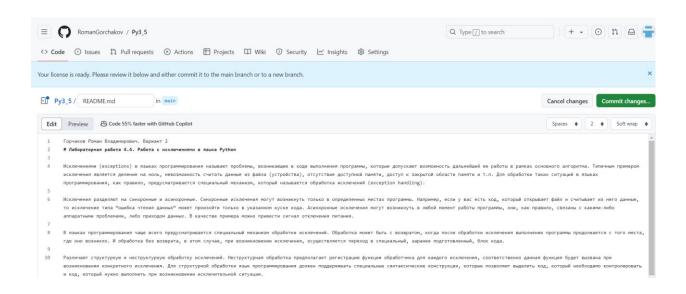


2. Теперь необходимо дополнить файл .gitignore с необходимыми правилами для языка программирования Python. Для этого переходим по ссылке «https://github.com/github/gitignore» и скачиваем оттуда файл «Python.gitignore».



```
# Byte-compiled / optimized / DLL files
       __pycache__/
 3
      *.py[cod]
      *$py.class
     # C extensions
      *.50
      # Distribution / packaging
10
      .Python
     build/
11
     develop-eggs/
      dist/
13
      downloads/
15
     eggs/
16
      .eggs/
      lib/
17
18
      lib64/
19
     parts/
20
     sdist/
21
      var/
      wheels/
23
     share/python-wheels/
24
     *.egg-info/
25
      .installed.cfg
26
      *.egg
     MANIFEST
27
28
```

3. Теперь создаём файл «README.md», где вносим ФИО и теоретический конспект лекции. Сохраняем набранный текст через кнопку «Commit changes».



4. В окне «Codespace» выбираем опцию «Create codespace on main». Откроется терминал, куда мы введём команду «git clone», чтобы клонировать свой репозиторий. После этого организуем репозиторий в соответствие с моделью ветвления Git-flow. Для этого введём в терминал команды: «git checkout —b develop» для создания ветки разработки; «git branch feature_branch» для создания ветки функций; «git branch release/1.0.0» для создания ветки релиза; «git checkout main» и «git branch hotfix» для создания веток hotfix. Устанавливаем библиотеки isort, black и flake8 и создаём файлы .pre-commit-config.yaml и environment.yml.

```
@RomanGorchakov →/workspaces/Py3_5 (main) $ git checkout -b develop
Switched to a new branch 'develop'

@RomanGorchakov →/workspaces/Py3_5 (develop) $ git branch feature_branch
@RomanGorchakov →/workspaces/Py3_5 (develop) $ git branch release/1.0.0

@RomanGorchakov →/workspaces/Py3_5 (develop) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

@RomanGorchakov →/workspaces/Py3_5 (main) $ git branch hotfix

@RomanGorchakov →/workspaces/Py3_5 (main) $ git checkout develop
Switched to branch 'develop'

@RomanGorchakov →/workspaces/Py3_5 (develop) $ []
```

```
Collecting black
   Downloading black-24.10.0-cp312-cp312-manylinux 2 17 x86 64.manylinux2014 x86 64.manylinux 2 28 x86 64.whl.metadata (79 kB)
  Collecting click>=8.0.0 (from black)
   Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
  Collecting mypy-extensions>=0.4.3 (from black)
   Downloading mypy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
  Requirement already satisfied: packaging>=22.0 in /home/codespace/.local/lib/python3.12/site-packages (from black) (24.1)
  Collecting pathspec>=0.9.0 (from black)
   Downloading pathspec-0.12.1-py3-none-any.whl.metadata (21 kB)
  Requirement already satisfied: platformdirs>=2 in /home/codespace/.local/lib/python3.12/site-packages (from black) (4.3.6)
  Downloading black-24.10.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl (1.8 MB)
                                             1.8/1.8 MB 51.8 MB/s eta 0:00:00
  Downloading click-8.1.7-py3-none-any.whl (97 kB)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
  Downloading pathspec-0.12.1-py3-none-any.whl (31 kB)
  Installing collected packages: pathspec, mypy-extensions, click, black
  Successfully installed black-24.10.0 click-8.1.7 mypy-extensions-1.0.0 pathspec-0.12.1
• @RomanGorchakov →/workspaces/Py3_5 (develop) $ pip install flake8
  Collecting flake8
   Downloading flake8-7.1.1-py2.py3-none-any.whl.metadata (3.8 kB)
  Collecting mccabe<0.8.0,>=0.7.0 (from flake8)
   Downloading mccabe-0.7.0-py2.py3-none-any.whl.metadata (5.0 kB)
  Collecting pycodestyle<2.13.0,>=2.12.0 (from flake8)
   Downloading pycodestyle-2.12.1-py2.py3-none-any.whl.metadata (4.5 kB)
  Collecting pyflakes<3.3.0,>=3.2.0 (from flake8)
   Downloading pyflakes-3.2.0-py2.py3-none-any.whl.metadata (3.5 kB)
  Downloading flake8-7.1.1-py2.py3-none-any.whl (57 kB)
  Downloading mccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
  Downloading pycodestyle-2.12.1-py2.py3-none-any.whl (31 kB)
  Downloading pyflakes-3.2.0-py2.py3-none-any.whl (62 kB)
  Installing collected packages: pyflakes, pycodestyle, mccabe, flake8
  Successfully installed flake8-7.1.1 mccabe-0.7.0 pycodestyle-2.12.1 pyflakes-3.2.0
• @RomanGorchakov →/workspaces/Py3_5 (develop) $ pre-commit sample-config > .pre-commit-config.yaml
• @RomanGorchakov →/workspaces/Py3_5 (develop) $ conda env export > environment.yml

@RomanGorchakov →/workspaces/Py3_5 (develop) $
```

5. Создаём файл «example.py», в котором нужно добавить возможность работы с исключениями и логгирование для примера 2 лабораторной работы 9.

Comm	and Prompt $ imes$	+ ~				-		×
>>> add Фамилия Должност Год пост >>> add Фамилия Должност Год пост >>> add Фамилия Должност Должност	и инициалы? Ивано ть? Директор тупления? 2007 и инициалы? Петро ть? Бухгалтер тупления? 2010 и инициалы? Сидор ть? Главный инжене тупления? 2012	в И.И. в П.П. ов С.С. р	риентированное програ		ример>руthon	exam	iple.	ру
N ₂	Ф.И.		+ Должность	Год				
1 1 2 1	Иванов И.И. Петров П.П. Сидоров С.С.		 Директор Бухгалтер Главный инженер	2007 2010 2012				
>>> exit	:		риентированное програг	+ ммирование\5\П	ример>			

6. Создаём файл «task1.py», в котором программа запрашивает ввод двух значений. Если хотя бы одно из них не является числом, то должна выполняться конкатенация, т. е. соединение, строк. В остальных случаях введенные числа суммируются.

```
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\5\Задания\Задание 1>python task1.py
Первое значение: 7.2
Второе значение: 2.6
Результат: 9.8
Все верно, вы ввели оба числа 7.2 и 2.6
Программа завершена.

C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\5\Задания\Задание 1>python task1.py
Первое значение: А
Второе значение: 113
Результат: A113
Программа завершена.

C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\5\Задания\Задания 1>
```

7. Создаём файл «task2.py», в котором программа генерировать матрицу из случайных целых чисел. Пользователь может указать число строк и столбцов, а также диапазон целых чисел. Производим обработку ошибок ввода пользователя.

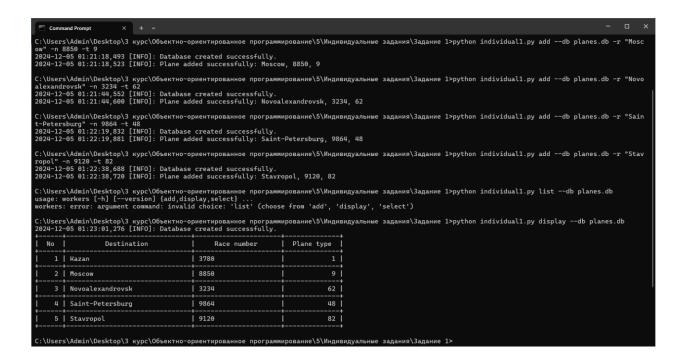
```
Command Prompt × + 

C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\5\Задания\Задание 2>python task2.py
Введите количество строк: 5
Введите количество стобцов: 5
Введите минимальное значение: 2
Введите максимальное значение: 10
Стенерированная матрица:
[4, 4, 2, 8, 5]
[7, 4, 9, 8, 6]
[3, 10, 10, 7, 3]
[6, 2, 10, 9, 9]
[7, 4, 7, 9, 4]

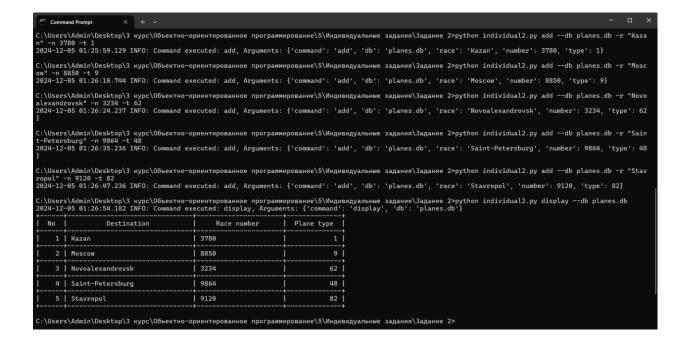
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\5\Задания\Задание 2>python task2.py
Введите количество строк: 4.5
Ошибка: Введите целое число.

C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\5\Задания\Задание 2>
```

8. Создаём файл «individual1.py», в котором нужно выполнить индивидуальное задание 1 лабораторной работы 2.19, добавив возможность работы с исключениями и логгирование.

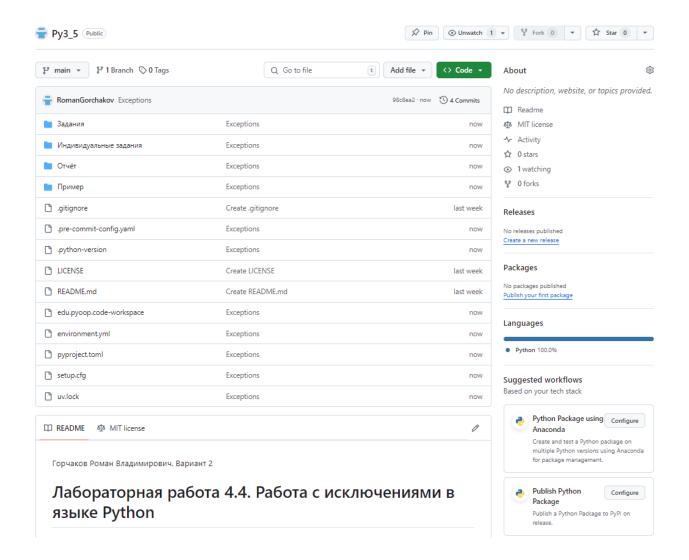


9. Создаём файл «individual2.py», в котором нужно добавить для предыдущего задания вывод в файлы лога даты и времени выполнения пользовательской команды с точностью до миллисекунды.



10. Выполняем коммит файлов в репозиторий Git в ветку разработки, сливаем её с веткой main и отправляем изменения на сервер GitHub.

```
16 files changed, 1359 insertions(+)
               create mode 100644 .pre-commit-config.yaml create mode 100644 .python-version
               create mode 100644 edu.pyoop.code-workspace
create mode 100644 environment.yml
               create mode 100644 pyproject.toml
create mode 100644 setup.cfg
             create mode 100644 w.lock create mode 100644 w.lock create mode 100644 "\320\227\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\2
          Create mode 100644 "\320\230\320\275\320\264\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\260\320\2
          20\260\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275
         20\260\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275\320\275
               create mode 100644 "\320\237\321\200\320\270\320\274\320\265\321\200/example.py"
         @RomanGorchakov →/workspaces/Py3_5 (main) $ git push -u
         Enumerating objects: 27, done.
Counting objects: 100% (27/27), done.
         Delta compression using up to 2 threads
Compressing objects: 100% (21/21), done.
         Writing objects: 100% (26/26), 867.37 KiB \mid 15.49 MiB/s, done. Total 26 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
         remote: Resolving deltas: 100% (2/2), done. To https://github.com/RomanGorchakov/Py3_5
         23452fe..98c8ea2 main -> main
branch 'main' set up to track 'origin/main'
○ @RomanGorchakov →/workspaces/Py3_5 (main) $
```



Контрольные вопросы

1. Какие существуют виды ошибок в языке программирования Python?

В Python выделяют два различных вида ошибок: синтаксические ошибки и исключения.

2. Как осуществляется обработка исключений в языке программирования Python?

Обработка исключений нужна для того, чтобы приложение не завершалось аварийно каждый раз, когда возникает исключение. Для этого блок кода, в котором возможно появление исключительной ситуации необходимо поместить во внутрь синтаксической конструкции try... except.

3. Для чего нужны блоки finally и else при обработке исключений?

Не зависимо от того, возникнет или нет во время выполнения кода в блоке try исключение, код в блоке finally все равно будет выполнен. Конструкция else, которая выполняется в случае, если в коде не произошло исключений.

- 4. Как осуществляется генерация исключений в языке Python? Для принудительной генерации исключения используется инструкция raise.
- 5. Как создаются классы пользовательский исключений в языке Python?
- В Python можно создавать собственные исключения. Такая практика позволяет увеличить гибкость процесса обработки ошибок в рамках той предметной области, для которой написана ваша программа. Для реализации собственного типа исключения необходимо создать класс, являющийся наследником от одного из классов исключений.
 - 6. Каково назначение модуля logging?

С помощью logging на Python можно записывать в лог и исключения. Обычно лог пишется в файл, зададим его как log.txt

7. Какие уровни логгирования поддерживаются модулем logging? Приведите примеры, в которых могут быть использованы сообщения с этим уровнем журналирования.

Самый низкий уровень логгирования — DEBUG. Он предназначен для отладочных сообщений и вывода диагностической информации о приложении. Уровень INFO указывает, что сообщения уровней ниже не будут отражаться в логе. Уровень WARNING предусматривает вывод предупреждений, он применяется для записи сведений о событиях, на которые программист обычно обращает внимание. Уровень ERROR предусматривает вывод сведений об ошибках. Уровень CRITICAL используется для вывода сведений об очень серьёзных ошибках, наличие которых угрожает нормальному функционированию всего приложения.

import logging
logging.basicConfig(level = logging.DEBUG)
logging.debug("Debug message!")
logging.info("Info message!")
logging.warning("Warning message!")
logging.error("Error message!")
logging.critical("Critical message!")

- >> DEBUG:root:Debug message!
 >> INFO:root:Info message!
 >> WARNING:root:Warning message!
 >> ERROR:root:Error message!
 >> CRITICAL:root:Critical message!