

ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ

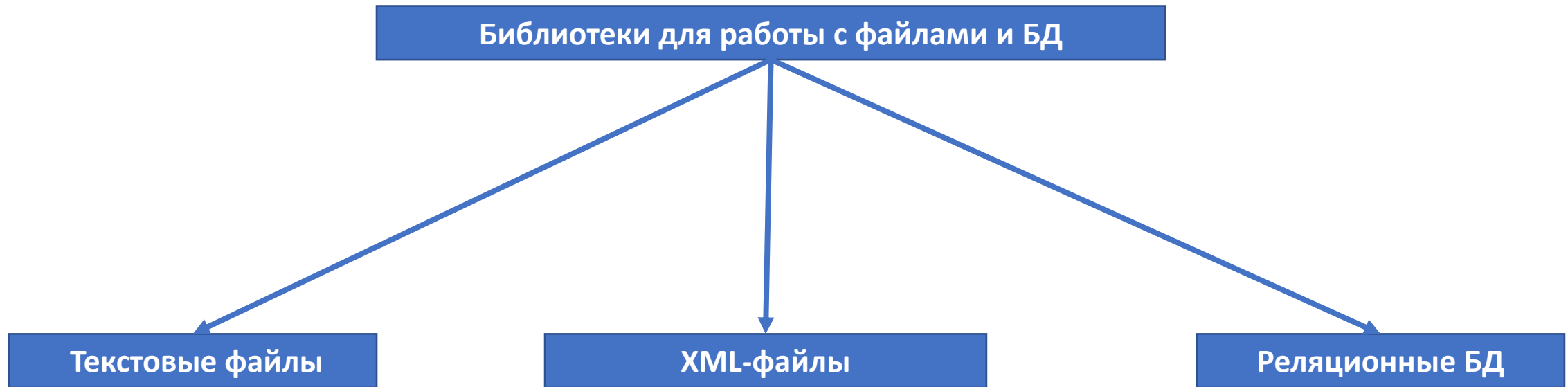
КАФЕДРА АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

ЛЕКЦИЯ №9

# Средства стандартной библиотеки языка программирования Python для работы с файлами и базой данных

СОСТАВИТЕЛЬ:            КАНД. ТЕХН. НАУК    БЫКАДОР В.С.

# Стандартная библиотека для работы с файлами и базой данных (БД)



# Работа с текстовыми файлами

Функция `open()` открывает файл и возвращает файловый объект. Режим `mode` указывает как необходимо открыть файл:

`mode = 'r'` – для чтения из файла;

`mode = 'w'` – для записи в файл.

После окончания работы с файлом необходимо вызвать метод `close()` !!!

Метод `readlines()` возвращает все строки из прочитанного файла!!!

```
example.txt
1  Привет!
2  Магистранты группы МИСБ11!
3  Как Ваши дела.
```

```
1
2  f = open('example.txt', mode='r')
3  for line in f.readlines():
4      print(line)
5
6  f.close()
7
```

```
• (venv) vityalybykador@Air-Vitaly test % /Users/vi
r/PROJECTS/test/for_lectures2.py
Привет!

Магистранты группы МИСБ11!

Как Ваши дела.
○ (venv) vityalybykador@Air-Vitaly test %
```

# Варианты значений режима открытия файлов `mode`

<https://docs-python.ru/tutorial/vstroennye-funktsii-interpretatora-python/funktsija-open/>

- r** - открывает файл только для чтения;
- w** - открыт для записи (перед записью файл будет очищен);
- a** - открыт для добавления в конец файла;
- +** - символ обновления (чтение + запись);
- t** - символ текстового режима;
- b** - символ двоичного режима;
- a** - открывает файл для добавления;
- a+** - открывает файл для добавления и чтения;
- и др.

# Работа с текстовыми файлами (чтение)

example.txt

```
1 Привет!  
2 Магистранты группы МИСБ11!  
3 Как Ваши дела.
```




```
• (venv) vityalybykador@Air-Vitaly test % /Users/vi  
r/PROJECTS/test/for_lectures2.py  
Привет!  
↕  
Магистранты группы МИСБ11!  
↕  
Как Ваши дела.  
○ (venv) vityalybykador@Air-Vitaly test %
```

The screenshot shows a Python IDE with a dark theme. The top toolbar includes a 'Run' button (a green play icon) and a dropdown menu currently showing 'Нет кон'. The file explorer on the left shows a project structure with a file named 'for\_lectures2.py'. The main editor window displays the code for 'for\_lectures2.py', which consists of seven lines: a blank line, opening a file 'example.txt' in read mode, a loop that reads lines and prints them, and closing the file. The fourth line, 'print(line)', is highlighted with a green background. On the left sidebar, the 'Variables' (ПЕРЕМЕННЫЕ) panel is open, showing the 'Locals' scope. It displays the variable 'f' as a '\_io.TextIOWrapper' object and the variable 'line' with the value 'Привет!\n'. The 'Globals' scope is also visible but empty.

```
for_lectures2.py — test  
main.py for_lectures2.py  
ПЕРЕМЕННЫЕ  
Locals  
> special variables  
> f: <_io.TextIOWrapper na...  
   line: 'Привет!\n'  
Globals  
1  
2 f = open('example.txt', mode='r')  
3 for line in f.readlines():  
4     print(line)  
5  
6 f.close()  
7
```

# Работа с текстовыми файлами (чтение)

```
1    
2  f = open('example.txt', mode='rb')  Режим чтения и двоичный режим.  
3  for line in f.readlines():  
4      print(line)  
5  
6  f.close()  
7
```

ПРОБЛЕМЫ

ВЫХОДНЫЕ ДАННЫЕ

КОНСОЛЬ ОТЛАДКИ

ТЕРМИНАЛ

JUPYTER

```
• (venv) vityalybykador@Air-Vitaly test % /Users/vitalybykador/PROJECTS/test/venv/bin/python /Users/vitalybykador/PROJECTS/test/for_lectures2.py  
b'\\xd0\\x9f\\xd1\\x80\\xd0\\xb8\\xd0\\xb2\\xd0\\xb5\\xd1\\x82!\\n'  
b'\\xd0\\x9c\\xd0\\xb0\\xd0\\xb3\\xd0\\xb8\\xd1\\x81\\xd1\\x82\\xd1\\x80\\xd0\\xb0\\xd0\\xbd\\xd1\\x82\\xd1\\x8b \\xd0\\xb3\\xd1\\x80\\xd1\\x83\\xd0\\xbf\\xd0\\xbf\\xd1\\x8b \\xd0\\x9c\\xd0\\x98\\xd0\\xa1\\xd0\\x9111!\\n'  
b'\\xd0\\x9a\\xd0\\xb0\\xd0\\xba \\xd0\\x92\\xd0\\xb0\\xd1\\x88\\xd0\\xb8 \\xd0\\xb4\\xd0\\xb5\\xd0\\xbb\\xd0\\xb0.'  
○ (venv) vityalybykador@Air-Vitaly test %
```

# Работа с текстовыми файлами (чтение)

```
1
2  f = open('example.txt', mode='r')
3  for line in f.readlines():
4      |   print(line.strip())
5
6  f.close()
7
```

ПРОБЛЕМЫ

ВЫХОДНЫЕ ДАННЫЕ

КОНСОЛЬ ОТЛАДКИ

- (venv) vityalybykador@Air-Vitaly test % /Users/vitaly  
r/PROJECTS/test/for\_lectures2.py

Привет!

Магистранты группы МИСБ11!

Как Ваши дела.

- (venv) vityalybykador@Air-Vitaly test % □

# Менеджер контекста

Использование менеджера контекста при работе с файлами позволяет гарантированно закрывать файлы даже в случае сбоя при чтении или записи в файл.

```
3 f = open('example.txt', mode='r')
4 try:
5     s = f.readline()
6     print(s)
7 finally:
8     f.close()
9
```



```
3 with open('example.txt', mode='r') as f:
4     s = f.readline()
5     print(s)
6
```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ

```
• (venv) vitalybykador@Air-Vitaly test % /Users/vitalybykador/PROJECTS/test/for_lectures2.py
Привет!
```

```
○ (venv) vitalybykador@Air-Vitaly test %
```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

```
• (venv) vitalybykador@Air-Vitaly test % /Users/vitalybykador/PROJECTS/test/for_lectures2.py
Привет!
```

```
○ (venv) vitalybykador@Air-Vitaly test %
```



# Работа с текстовыми файлами (запись)

```
3  s = '''
4  Это новые данные, которые
5  будут записаны в файл.
6  '''
7
8  with open('example.txt', mode='w') as f:
9      f.write(s)
10
```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

```
• (venv) vityalybykador@Air-Vitaly test % /Users/vitalybykador/P
r/PROJECTS/test/for_lectures2.py
○ (venv) vityalybykador@Air-Vitaly test %
```



example.txt

```
1
2  Это новые данные, которые
3  будут записаны в файл.
```

# Работа с текстовыми файлами (запись)

```
3 s = 'Это новые данные, которые будут записаны в файл.'  
4  
5 with open('example.txt', mode='w') as f:  
6     f.write(s)  
7
```




```
example.txt  
1 Это новые данные, которые будут записаны в файл.
```

# Работа с текстовыми файлами (запись)

```
2
3 s = '\nЭто новые данные, которые\nбудут записаны в файл.'
4
5 with open('example.txt', mode='w') as f:
6     | f.write(s)
7
```



 example.txt

```
1
2 Это новые данные, которые
3 будут записаны в файл.
```

## Работа с текстовыми файлами (запись)


```
2  💡  
3  s = 'Добавим пару слов.'  
4  
5  with open('example.txt', mode='a') as f:  
6  |    f.write(s)  
7
```



```
1  
2  Это новые данные, которые  
3  будут записаны в файл. Добавим пару слов.
```



## Работа с текстовыми файлами (запись списка строк)

```
2  lst = ['Толик\n', 'Ира\n', 'Света\n']
3
4  with open('example.txt', mode='a') as f:
5      for s in lst:
6          f.write(s)
7
```

 example.txt  
1 Толик  
2 Ира  
3 Света  
4

## Работа с текстовыми файлами (запись списка строк)

```
2 lst = ['Ира\n', 'Толик\n', 'Света\n']
3
4 with open('example.txt', mode='a') as f:
5     f.writelines(lst)
6
```

 example.txt  
1 Ира  
2 Толик  
3 Света  
4 

# Работа с текстовыми файлами (можно сделать свой формат файла)

```
1 class Person:
2     def __init__(self, name, age, job):
3         self.__name = name
4         self.__age = age
5         self.__job = job
6
7     def get_all_data(self):
8         return [self.__name, self.__age, self.__job]
9
10    def __str__(self):
11        return f'name: {self.__name}; age: {self.__age}; job: {self.__job}'
```

# Работа с текстовыми файлами (можно сделать свой формат файла)

```
1  from clsPerson import Person
2
3  person1 = Person('Толик', 26, 'Программист')
4  person2 = Person('Ира', 25, 'Переводчик')
5
6  print(person1)
7  print(person2)
8
```

ПРОБЛЕМЫ

ВЫХОДНЫЕ ДАННЫЕ

КОНСОЛЬ ОТЛАДКИ

ТЕРМИНАЛ

JS

```
• (venv) vitalitybykador@Air-Vitaly test % /Users/vitalybykador/PROJECTS/test/for_lectures2.py
name: Толик; age: 26; job: Программист
name: Ира; age: 25; job: Переводчик
○ (venv) vitalitybykador@Air-Vitaly test %
```



# Работа с текстовыми файлами (можно сделать свой формат файла)

```
1  from clsPerson import Person
2
3
4  def write_to_file(file_name, person):
5      with open(file_name, mode='a') as file:
6          data = person.get_all_data()
7          file.write(f'name: {data[0]};age: {data[1]};job: {data[2]}\n')
8
9  person1 = Person('Толик', 26, 'Программист')
10 person2 = Person('Ира', 25, 'Переводчик')
11
12 write_to_file('persons.dat', person1)
13 write_to_file('persons.dat', person2)
14
```

3

1

2

# Работа с текстовыми файлами

(записываем данные из класса в файл – самодельная сериализация)

```
persons.dat
1  name: Толик;age: 26;job: Программист
2  name: Ира;age: 25;job: Переводчик
3
```

# Работа с текстовыми файлами

(восстанавливаем  
данные из файла в  
класс —  
самодельная  
сериализация)

```
1  from clsPerson import Person
2
3
4  lst_persons = list()
5
6  with open('persons.dat', mode='r') as file:
7      for s in file.readlines():
8          name = s.split(';')[0].split(': ')[1].strip()
9          age = s.split(';')[1].split(': ')[1].strip()
10         job = s.split(';')[2].split(': ')[1].strip()
11         lst_persons.append(Person(name, age, job))
12
13     print(lst_persons[0])
14     print(lst_persons[1])
15
```

ПРОБЛЕМЫ

ВЫХОДНЫЕ ДАННЫЕ

КОНСОЛЬ ОТЛАДКИ

ТЕРМИНАЛ

JUPYTER

```
• (venv) vitalybykador@Air-Vitaly test % /Users/vitalybykador/PROJECTS/test/venv/bin/
  _lectures.py
  name: Толик; age: 26; job: Программист
  name: Ира; age: 25; job: Переводчик
○ (venv) vitalybykador@Air-Vitaly test %
```

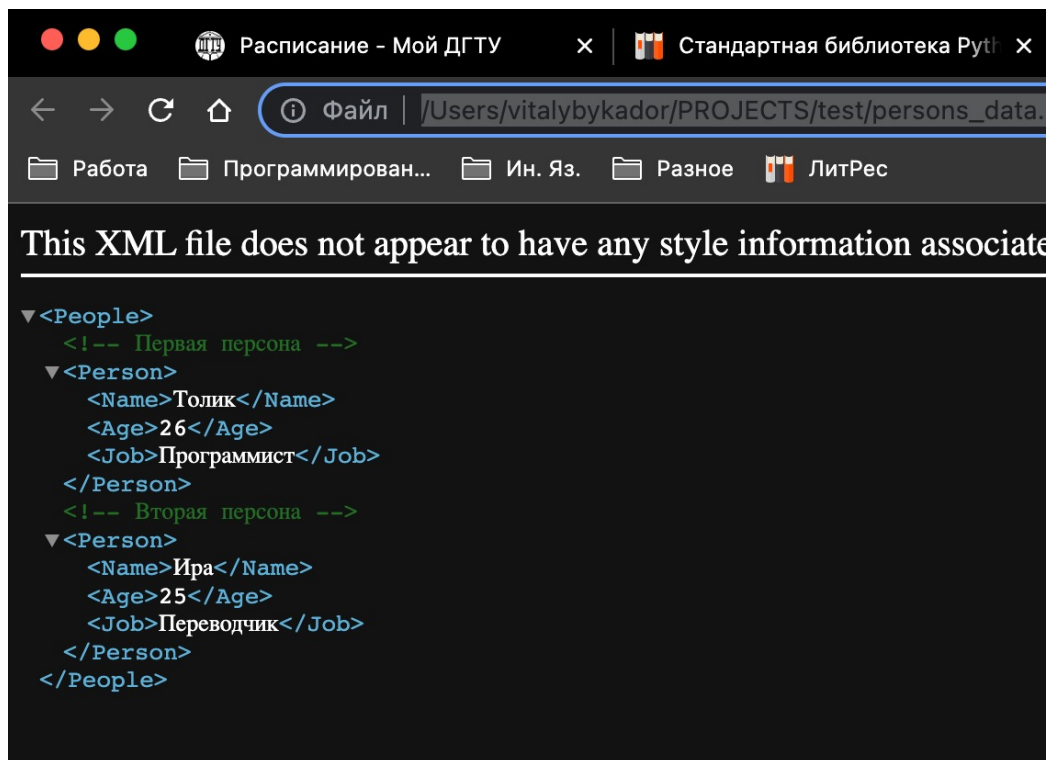
# XML-файлы для структурированного хранения информации

В XML можно создавать любые теги и делать любую вложенность тегов один в другой.

```
persons.dat
1  name: Толик;age: 26;job: Программист
2  name: Ира;age: 25;job: Переводчик
3
```



```
</> persons_data.xml
1  <?xml version="1.0" encoding = "UTF-8"?>
2  <!--Корневой узел обязателен!!!-->
3  <People>
4      <!--Первая персона-->
5      <Person>
6          <Name>Толик</Name>
7          <Age>26</Age>
8          <Job>Программист</Job>
9      </Person>
10
11     <!--Вторая персона-->
12     <Person>
13         <Name>Ира</Name>
14         <Age>25</Age>
15         <Job>Переводчик</Job>
16     </Person>
17 </People>
```



# Работа с XML-файлами

for\_lectures.py > ...

```
1  from xml.etree import ElementTree
2  from xml.etree import ElementTree
3
4  with open('persons_data.xml', mode='r') as f:
5      tree = ElementTree.parse(f)
6
7  for element in tree.iter():
8      print(element)
9
```

ПРОБЛЕМЫ

ВЫХОДНЫЕ ДАННЫЕ

КОНСОЛЬ ОТЛАДКИ

ТЕРМИНАЛ

JUPYTER

- (venv) vitalitybykador@Air-Vitaly test % /Users/vitalybykador/PROJECTS/for\_lectures.py  
<Element 'People' at 0x100688900>  
<Element 'Person' at 0x100688950>  
<Element 'Name' at 0x1006889a0>  
<Element 'Age' at 0x1006889f0>  
<Element 'Job' at 0x100688a40>  
<Element 'Person' at 0x100688a90>  
<Element 'Name' at 0x100688ae0>  
<Element 'Age' at 0x100688b30>  
<Element 'Job' at 0x100688b80>
- (venv) vitalitybykador@Air-Vitaly test %

# Работа с XML-файлами (чтение по элементам)

```
</> persons_data.xml
1  <?xml version="1.0" encoding = "UTF-8"?>
2  <!--Корневой узел обязателен!!!-->
3  <People>
4      <!--Первая персона-->
5      <Person>
6          <Name>Толик</Name>
7          <Age>26</Age>
8          <Job>Программист</Job>
9      </Person>
10
11     <!--Вторая персона-->
12     <Person>
13         <Name>Ира</Name>
14         <Age>25</Age>
15         <Job>Переводчик</Job>
16     </Person>
17 </People>
```

```
1  from xml.etree import ElementTree
2
3
4  tree = ElementTree.parse('persons_data.xml')
5  root = tree.getroot()
6
7  for node in root.iter('Person'):
8      print()
9      print(node.findtext('Name'))
10     print(node.findtext('Age'))
11     print(node.findtext('Job'))
12
```

ПРОБЛЕМЫ	ВЫХОДНЫЕ ДАННЫЕ	КОНСОЛЬ ОТЛАДКИ	ТЕРМИНАЛ	JU
●	(venv) vitalitybykador@Air-Vitaly test % /Users/vitalybykador/PROJECT res.py		Толик 26 Программист  Ира 25 Переводчик	
○	(venv) vitalitybykador@Air-Vitaly test %			

# Работа с XML-файлами (добавление элемента в файл)

```
1  from xml.etree import ElementTree
2
3
4  tree = ElementTree.parse('persons_data.xml')
5  root = tree.getroot()
6
7  person = ElementTree.Element('Person')
8  name = ElementTree.SubElement(person, 'Name').text = 'Гоша'
9  age = ElementTree.SubElement(person, 'Age').text = '20'
10 job = ElementTree.SubElement(person, 'Job').text = 'Официант'
11
12 root.append(person)
13
14 tree.write('persons_data.xml', encoding='utf8')
15
```

# Работа с XML-файлами (добавление элемента в файл)

</> persons\_data.xml

```
1  <?xml version='1.0' encoding='utf8'?>
2  <People>
3
4      <Person>
5          <Name>Толик</Name>
6          <Age>26</Age>
7          <Job>Программист</Job>
8      </Person>
9
10
11     <Person>
12         <Name>Ира</Name>
13         <Age>25</Age>
14         <Job>Переводчик</Job>
15     </Person>
16 <Person><Name>Гоша</Name><Age>20</Age><Job>Официант</Job></Person></People>
```



# Работа с XML-файлами (вывод по конкретным тегам)

```
1 from xml.etree import ElementTree
2
3 parser = ElementTree.XMLParser(encoding='utf-8')
4 tree = ElementTree.parse('persons_data.xml', parser)
5 root = tree.getroot()
6
7 for node in root.findall('./Person/Name'):
8     print(node.text)
9
```

Была ошибка в кодировке после записи. Пришлось указать кодировку xml-парсеру.

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    JUPYTER

```
• (venv) vitallybykador@MacBook-Air-Vitaly test % /Users/vitalybykador/PROJECTS
ECTS/test/for_lectures.py
Толик
Ира
Гоша
```

```
</> persons_data.xml
1 <?xml version='1.0' encoding='utf8'?>
2 <People>
3     <Person>
4         <Name>Толик</Name>
5         <Age>26</Age>
6         <Job>Программист</Job>
7     </Person>
8     <Person>
9         <Name>Ира</Name>
10        <Age>25</Age>
11        <Job>Переводчик</Job>
12    </Person>
13    <Person>
14        <Name>Гоша</Name>
15        <Age>20</Age>
16        <Job>Официант</Job>
17    </Person>
18 </People>
```

# Работа с XML-файлами (вывод по конкретным тегам)

```
1 from xml.etree import ElementTree
2
3 parser = ElementTree.XMLParser(encoding='utf-8')
4 tree = ElementTree.parse('persons_data.xml', parser)
5 root = tree.getroot()
6
7 for node in root.findall('./Person/Job'):
8     print(node.text)
9
```

**XPath** - синтаксис. Тоже типа микроязыка наподобие регулярных выражений.

<https://docs.python.org/3/library/xml.etree.elementtree.html#xpath-support>

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    JUPYTER

```
• (venv) vitalitybykador@MacBook-Air-Vitaly test % /Users/vitalybykador/PROJECTS/test/for_lectures.py
Толик
Ира
Гоша
• (venv) vitalitybykador@MacBook-Air-Vitaly test % /Users/vitalybykador/PROJECTS/test/for_lectures.py
Программист
Переводчик
Официант
• (venv) vitalitybykador@MacBook-Air-Vitaly test %
```

```
<?xml version='1.0' encoding='utf8'?>
<People>
  <Person>
    <Name>Толик</Name>
    <Age>26</Age>
    <Job>Программист</Job>
  </Person>
  <Person>
    <Name>Ира</Name>
    <Age>25</Age>
    <Job>Переводчик</Job>
  </Person>
  <Person>
    <Name>Гоша</Name>
    <Age>20</Age>
    <Job>Официант</Job>
  </Person>
</People>
```

# Работа с реляционными базами данных

ПРОВОДНИК

TEST

- index.html
- main.py
- main2.py
- persons\_data.xml
- persons.dat
- ToolMachines.db

СТРУКТУРА

ВРЕМЕННАЯ ШКАЛА

SQLITE EXPLORER

- ToolMachines.db
  - dicGroupToolMachines
  - dicInvNumToolMachines
  - dicManagers
  - dicModelToolMachine
  - dicServiceWorks
  - dicShops
  - dicToolMachinesShops
  - factServices

InvNumber	IdGroup	IdModel
16236	1	2
44127	2	12
50050	2	28
50624	3	18
73309	2	24
77547	6	8
106515	5	22
112104	2	14
114995	8	34
130720	3	15
135331	2	25
135501	6	7
140109	3	17
152248	2	24

Id	Name
1	Токарный
2	Сверлильный
3	Шлифовальный
5	Зубообрабатывающий
6	Фрезерный
7	Протяжной
8	Отрезной

Id	Name
1	ИЖ 250
2	1A616
3	16K20
4	ДИП 300
5	16M30ФЗ
6	1516ФЗ
7	6Н10
8	6Т80Ш
9	ВМ127
10	ГФ2171С5
11	65А90

# Работа с реляционными базами данных


test\_SQL.sql

```
1  -- SQLite
2  SELECT
3      dicInvNumToolMachines.InvNumber,
4      dicGroupToolMachines.Name,
5      dicModelToolMachine.Name
6  FROM dicInvNumToolMachines
7      INNER JOIN dicGroupToolMachines
8          ON dicInvNumToolMachines.IdGroup = dicGroupToolMachines.Id
9      INNER JOIN dicModelToolMachine
10         ON dicInvNumToolMachines.IdModel = dicModelToolMachine.Id
11
```

InvNumber	Name	Name
16236	Токарный	1A616
44127	Сверлильный	2421
50050	Сверлильный	2Н115ПМ
50624	Шлифовальный	ЗД725
73309	Сверлильный	НС-12А
77547	Фрезерный	6Т80Ш
106515	Зубообрабатывающий	5В12
112104	Сверлильный	2Е470
114995	Отрезной	ВШ-042
130720	Шлифовальный	ЗГ71
135331	Сверлильный	2М112
135501	Фрезерный	6Н10
140109	Шлифовальный	ЗБ722
153248	Сверлильный	НС-12А
172038	Фрезерный	6Т80Ш
191116	Токарный	1516Ф3
192245	Шлифовальный	ЗБ725

# Работа с реляционными базами данных


```
2 SELECT
3     dicInvNumToolMachines.InvNumber,
4     dicGroupToolMachines.Name,
5     dicModelToolMachine.Name
6 FROM dicInvNumToolMachines
7     INNER JOIN dicGroupToolMachines
8         ON dicInvNumToolMachines.IdGroup = dicGroupToolMachines.Id
9     INNER JOIN dicModelToolMachine
10        ON dicInvNumToolMachines.IdModel = dicModelToolMachine.Id
11
12 WHERE
13     dicInvNumToolMachines.InvNumber = 77547
```



77547	6	8
106515	5	22
112104	2	14
114995	8	34
130720	3	15
135331	2	25
135501	6	7
140109	3	17

Id	Name
1	Токарный
2	Сверлильный
3	Шлифовальный
5	Зубообрабатывающий
6	Фрезерный

3	10K20
4	ДИП 300
5	16M30Ф3
6	1516Ф3
7	6H10
8	6T80Ш



InvNumber	Name	Name
77547	Фрезерный	6T80Ш

# Работа с реляционными базами данных

```
1 import sqlite3
2
3 with sqlite3.connect('ToolMachines.db') as connection:
4     cursor = connection.cursor()
5
6     cursor.execute("""
7         SELECT
8             dicInvNumToolMachines.InvNumber,
9             dicGroupToolMachines.Name,
10            dicModelToolMachine.Name
11        FROM dicInvNumToolMachines
12            INNER JOIN dicGroupToolMachines
13                ON dicInvNumToolMachines.IdGroup = dicGroupToolMachines.Id
14            INNER JOIN dicModelToolMachine
15                ON dicInvNumToolMachines.IdModel = dicModelToolMachine.Id
16        """)
17
18 for row in cursor.fetchall():
19     inv_num, group, model = row
20     print(inv_num, group, model)
```

Создать подключение к БД.

Создать КУРСОР.

Выполнить SQL-запрос к БД.

Получить каждую отдельную строку в из результирующего набора данных.

# Работа с реляционными базами данных

```
6 cursor.execute("""
7 SELECT
```

	ПРОБЛЕМЫ	ВЫХОДНЫЕ ДАННЫЕ	КОНСОЛЬ ОТЛ
702958	Шлифовальный	ЗД711	
718476	Сверлильный	НС-12А	
718696	Токарный	16М30Ф3	
745243	Фрезерный	ВМ127	
751538	Шлифовальный	ЗГ71	
760035	Зубообрабатывающий	53А30	
766606	Фрезерный	ВМ127	
770792	Токарный	ИЖ 250	
777969	Токарный	1А616	
778983	Сверлильный	2Н115ПМ	
791280	Сверлильный	2А106П	
805883	Зубообрабатывающий	5310	
814606	Протяжной	7А734	
821818	Зубообрабатывающий	5М14	
845585	Отрезной	ОН 401	
845658	Отрезной	ВШ-042	
869579	Шлифовальный	ЗБ722	
877996	Сверлильный	2М112	
888702	Токарный	1516Ф3	
893737	Зубообрабатывающий	5М14	
899714	Сверлильный	ГС2112	
916954	Сверлильный	2Е470	
928791	Шлифовальный	ЗД711	
934053	Фрезерный	6Н10	
936405	Токарный	1516Ф3	
936566	Отрезной	ОН 401	
945432	Фрезерный	6Н10	



# Работа с реляционными базами данных

```
for_lectures.py > ...
14         INNER JOIN dicModelToolMachine
15             ON dicInvNumToolMachines.IdModel = dicModelToolMachine.Id
16         """)
17
18     for row in cursor.fetchall():
19         inv_num, group, model = row
20         print('| {:^10}| {:<20}| {:<10}|'.format(inv_num, group, model))
```

ПРОБЛЕМЫ

ВЫХОДНЫЕ ДАННЫЕ

КОНСОЛЬ ОТЛАДКИ

ТЕРМИНАЛ

JUPYTER: VARIABLES

718476	Сверлильный	НС-12А
718696	Токарный	16М30Ф3
745243	Фрезерный	ВМ127
751538	Шлифовальный	ЗГ71
760035	Зубообрабатывающий	53А30
766606	Фрезерный	ВМ127
770792	Токарный	ИЖ 250
777969	Токарный	1А616
778983	Сверлильный	2Н115ПМ
791280	Сверлильный	2А106П
805883	Зубообрабатывающий	5310
814606	Протяжной	7А734
821818	Зубообрабатывающий	5М14
845585	Отрезной	ОН 401
845658	Отрезной	ВШ-042
869579	Шлифовальный	ЗБ722
877996	Сверлильный	2М112
888702	Токарный	1516Ф3
893737	Зубообрабатывающий	5М14
899714	Сверлильный	ГС2112
916954	Сверлильный	2Е470
928791	Шлифовальный	ЗД711
934053	Фрезерный	6Н10
936405	Токарный	1516Ф3
936566	Отрезной	ОН 401
945432	Фрезерный	6Н10
946125	Протяжной	7А523
953470	Шлифовальный	ЗГ71
961816	Фрезерный	ВМ127
975542	Шлифовальный	ЗГ71
990302	Отрезной	ОН 401




# Работа с реляционными базами данных

```
14         INNER JOIN dicModelToolMachine
15         ON dicInvNumToolMachines.IdModel = dicModelToolMachine.Id
16     """
17
18     with open('data_from_db.txt', mode='w') as f:
19         for row in cursor.fetchall():
20             inv_num, group, model = row
21             f.write('| {:^10}| {:<20}| {:<10}| \n'.format(inv_num, group, model))
22
```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    JUPYTER: VARIABLES

```
(venv) vityalybykador@MacBook-Air-Vitaly test % /Users/vitalybykador/PROJECTS/test/venv/bin/python /Users/vita
dor/PROJECTS/test/for_lectures.py
(venv) vityalybykador@MacBook-Air-Vitaly test %
```



data\_from\_db.txt

1	16236	Токарный	1A616
2	44127	Сверлильный	2421
3	50050	Сверлильный	2Н115ПМ
4	50624	Шлифовальный	ЗД725
5	73309	Сверлильный	НС-12А
6	77547	Фрезерный	6Т80Ш
7	106515	Зубообрабатывающий	5В12
8	112104	Сверлильный	2Е470
9	114995	Отрезной	ВШ-042
10	130720	Шлифовальный	ЗГ71
11	135331	Сверлильный	2М112
12	135501	Фрезерный	6Н10
13	140109	Шлифовальный	ЗБ722
14	153248	Сверлильный	НС-12А
15	172038	Фрезерный	6Т80Ш
16	191116	Токарный	1516ФЗ
17	192345	Шлифовальный	ЗД725
18	195146	Токарный	16М30ФЗ
19	202210	Зубообрабатывающий	5В12
20	207580	Зубообрабатывающий	5310
21	243031	Токарный	16М30ФЗ
22	246031	Протяжной	7А734
23	246279	Токарный	1516ФЗ
24	246552	Отрезной	ОН 401

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    JUPYTER: VARIABLES

```
(venv) vityalybykador@MacBook-Air-Vitaly test % /Users/vitalybykador/PROJECTS/
dor/PROJECTS/test/for_lectures.py
(venv) vityalybykador@MacBook-Air-Vitaly test %
```