

ФЕДЕРАЛЬНОЕ АГЕНСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ
СООБЩЕНИЯ Императора Александра I»

Кафедра «Информационные и вычислительные системы»

Дисциплина «Программирование на языках высокого уровня (Python)»

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10

Выполнил студент
Факультет: АИТ
Группа: ИВБ-211

Шефнер А.

Проверил:

Баталов Д.И.

Санкт-Петербург

2023

Оценочный лист результатов ЛР № 10

Ф.И.О. студента _____ Шефнер Альберт _____

Группа _____ ИВБ-211 _____

№ п/ п	Материалы необходимые для оценки знаний, умений и навыков	Показатель оценивания	Критерии Оценивания	Шкала оценивания	Оценка
1	Лабораторная работа №	Соответствие методике выполнения	Соответствует	7	
			Не соответствует	0	
		Срок выполнения	Выполнена в срок	2	
			Выполнена с опозданием на 2 недели	0	
		оформление	Соответствует требованиям	1 0	
			Не соответствует		
	ИТОГО количество баллов			10	

Доцент кафедры

«Информационные и вычислительные
системы»

_____ 2023 г.

Баталов Д.И. «__»

4. Спроектируйте базу данных и напишите программу «Электронный каталог продуктов магазина» с возможностью сохранения, удаления и редактирования данных.

Код

shop.py — отправная точка приложения

```
1 import os
2 import platform
3 import sys
4 from collections import namedtuple
5
6 import shopdb as db
7 from tabulate import tabulate
8
9 Action = namedtuple("Action", ("desc", "func"))
10
11 id_fields = ["id"] + db.fields
12 human_id_fields = ["ID"] + db.human_fields
13
14 clear = ""
15 match platform.system():
16     case "Linux" | "Darwin" | "Unix":
17         clear = "clear"
18     case "Windows":
19         clear = "cls"
20
21
22 def get_field_from_user():
23     print("Введите номер поля:")
24     for number, field in enumerate(human_id_fields):
25         print(f"{number} - {field}")
26     field_num = int(input())
27     field = id_fields[field_num]
28
29     value = input("Введите значение поля: ")
30
31     return field, value
32
33
34 def exit():
35     global running
36     running = False
37
38
39 def get_all_records():
40     print(tabulate(db.get_all_records(), human_id_fields))
```

```
43 def get_records_by_field():
44     field, value = get_field_from_user()
45     print(tabulate(db.get_records_by_field(field, value), id_fields))
46
47
48 def insert_record():
49     values = (input(f"Введите '{field}': ") for field in db.human_fields)
50     db.insert_record(db.Record(*values))
51
52
53 def update_field():
54     id = input("Введите ID записи: ")
55     field, value = get_field_from_user()
56     db.update_field(id, field, value)
57
58
59 def delete():
60     id = int(input("Введите ID: "))
61     db.delete(id)
62
63
64 def delete_by_field():
65     field, value = get_field_from_user()
66     db.delete_by_field(field, value)
67
68
69 actions = [
70     Action("выйти из программы", exit),
71     Action("вывести все записи", get_all_records),
72     Action("вывести значения по полю", get_records_by_field),
73     Action("добавить новую запись", insert_record),
74     Action("обновить запись по ID", update_field),
75     Action("удалить запись по ID", delete),
76     Action("удалить запись по полю", delete_by_field),
77 ]
78
79
80 if __name__ == "__main__":
81     try:
82         db.create_table()
83
84         running = True
85         while running:
86             os.system(clear)
87
88             print("Введите номер желаемого действия:")
89             for number, action in enumerate(actions):
90                 print(f"{number} - {action.desc}")
```

```

91
92     try:
93         action_number = int(input())
94         actions[action_number].func()
95     except IndexError:
96         print("Введённый номер был вне границ массива")
97     except ValueError:
98         print("Ошибка входного значения")
99     except KeyboardInterrupt:
100         raise KeyboardInterrupt
101     except Exception as e:
102         print("Ошибка:", e)
103
104         input("Нажмите ENTER для продолжения...")
105
106 except KeyboardInterrupt:
107     sys.exit(0)

```

shopdb.py — функции работы с базой данных

```

1 import sqlite3
2 from collections import namedtuple
3
4 record_name = "product"
5 table_name = "products"
6 db_file = f"{table_name}.db"
7 fields = ["name", "unit", "amount", "price"]
8 field_db_types = ["TEXT", "TEXT", "INTEGER", "INTEGER"]
9
10 human_fields = ["Название", "Единица измерения", "Количество в упаковке", "Цена"]
11 human_record_name = "Продукт"
12
13 Record = namedtuple(record_name, fields)
14
15 conn = sqlite3.connect(db_file)
16 curs = conn.cursor()
17
18
19 def create_table():
20     sql = f"CREATE TABLE IF NOT EXISTS {table_name} (id INTEGER PRIMARY KEY AUTOINCREMENT, "
21     sql += ", ".join(map(lambda t: f"{t[1]} {t[0]}", zip(field_db_types, fields)))
22     sql += ")"
23     curs.execute(sql)
24     conn.commit()
25
26
27 def get_all_records():
28     curs.execute(f"SELECT * FROM {table_name}")
29     return curs.fetchall()

```



```

32 def get_records_by_field(field, value):
33     curs.execute(f"SELECT * FROM {table_name} WHERE {field}=?", (value,))
34     return curs.fetchall()
35
36
37 def insert_record(record):
38     with conn:
39         curs.execute(
40             f"INSERT INTO {table_name} VALUES (NULL, {'', '}.join('? ' * len(fields)))",
41             record,
42         )
43
44
45 def update_field(id, field, value):
46     with conn:
47         curs.execute(f"UPDATE {table_name} SET {field}=? WHERE id=?", (value, id))
48
49
50 def delete(id):
51     with conn:
52         curs.execute(f"DELETE FROM {table_name} WHERE id=?", (id,))
53
54
55 def delete_by_field(field, value):
56     with conn:
57         curs.execute(f"DELETE FROM {table_name} WHERE {field}=?", (value,))

```

Отладка

```

Введите номер желаемого действия:
0 - выйти из программы
1 - вывести все записи
2 - вывести значения по полю
3 - добавить новую запись
4 - обновить запись по ID
5 - удалить запись по ID
6 - удалить запись по полю

```

Введите номер желаемого действия:

- 0 - выйти из программы
 - 1 - вывести все записи
 - 2 - вывести значения по полю
 - 3 - добавить новую запись
 - 4 - обновить запись по ID
 - 5 - удалить запись по ID
 - 6 - удалить запись по полю
- 3

введите 'Название': Яблоко

введите 'Единица измерения': килограмм

введите 'Количество в упаковке': 1

введите 'Цена': 90

Нажмите ENTER для продолжения...

Введите номер желаемого действия:

- 0 - выйти из программы
 - 1 - вывести все записи
 - 2 - вывести значения по полю
 - 3 - добавить новую запись
 - 4 - обновить запись по ID
 - 5 - удалить запись по ID
 - 6 - удалить запись по полю
- 1

ID	Название	Единица измерения	Количество в упаковке	Цена
1	Гречка	грамм	800	100
2	Яблоко	килограмм	1	90
3	Молоко	литр	1	150
5	Сухарики	грамм	150	50

Нажмите ENTER для продолжения...

9. Спроектируйте базу данных и напишите программу «Электронная телефонная книга» с возможностью сохранения, удаления и редактирования данных.

Код

Отличается лишь набором констант в файле исходного кода функций базы данных.

```
4 record_name = "contact"
5 table_name = "contacts"
6 db_file = f"{table_name}.db"
7 fields = ["first", "last", "phone"]
8 field_db_types = ["TEXT", "TEXT", "TEXT"]
9
10 human_fields = ["Имя", "Фамилия", "Номер телефона"]
11 human_record_name = "Контакт"
```

Отладка

```
Введите номер желаемого действия:
0 - выйти из программы
1 - вывести все записи
2 - вывести значения по полю
3 - добавить новую запись
4 - обновить запись по ID
5 - удалить запись по ID
6 - удалить запись по полю
1
  ID  Имя    Фамилия      Номер телефона
----  -
   1  Борис  Чайковский   88005553535
Нажмите ENTER для продолжения...
```