**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторным работам №1 - 10**

**по дисциплине «Объектно-ориентированное программирование на языке Java»**

|  |  |
| --- | --- |
| Студент гр. 7307 | Торопов В. А. |
| Преподаватель | Павловский М.Г. |

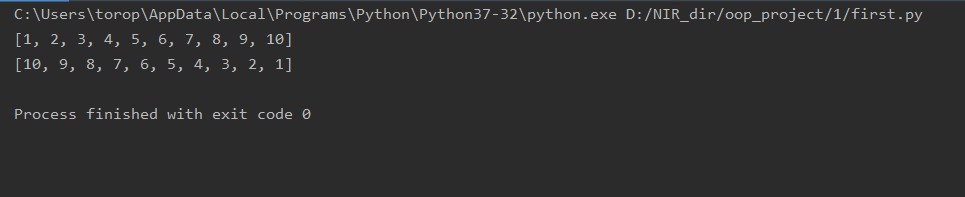
Санкт-Петербург

2019

# Лабораторная работа № 1

Цель: Освоение среды разработки Eclipse, программирование, запуск и отладка консольного приложения.

Проверка работоспособности:



Текст программы:

**"""first lab"""**

**import random**

**def quicksort(nums):**

**"""function of sorting"""**

**if len(nums) <= 1:**

**return nums**

**else:**

**q = random.choice(nums)**

**l\_nums = [n for n in nums if n > q]**

**e\_nums = [q] \* nums.count(q)**

**b\_nums = [n for n in nums if n < q]**

**return quicksort(l\_nums) + e\_nums + quicksort(b\_nums)**

**def main():**

**"""main of program"""**

**mas = [i for i in range(1, 11)]**

**print(mas)**

**mas = quicksort(mas)**

**print(mas)**

**if \_\_name\_\_ == "\_\_main\_\_":**

**main()**

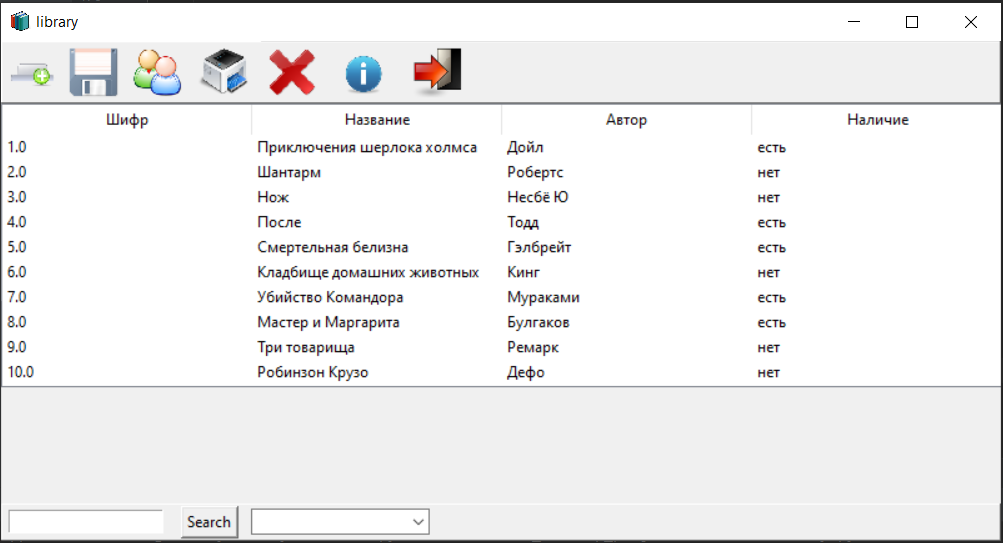
# Лабораторная работа № 2 - 5

Цель:

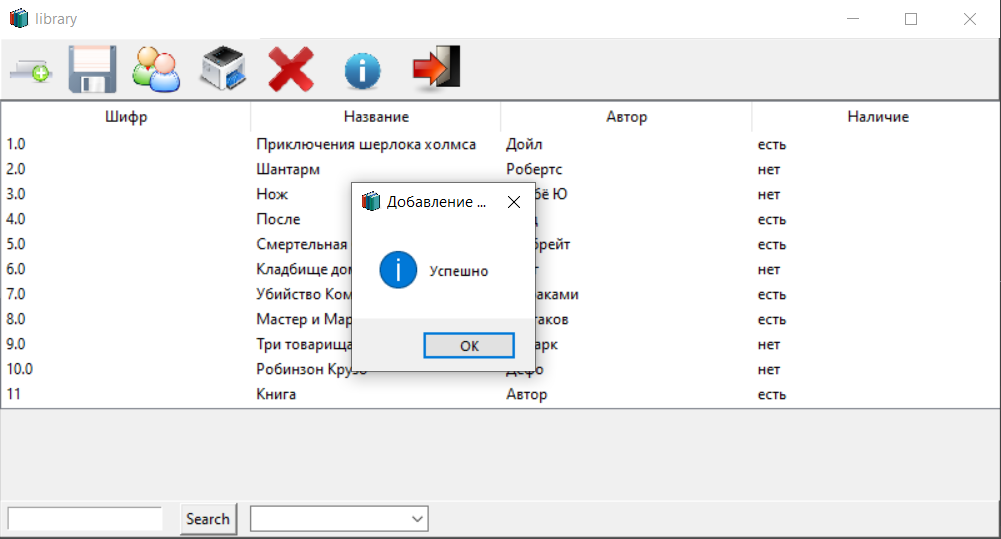
1. Знакомство с правилами построения экранной формы.
2. Обработка событий.
3. Обработка исключений.
4. Сохранение и загрузка данных из файла.

Проверка работоспособности:

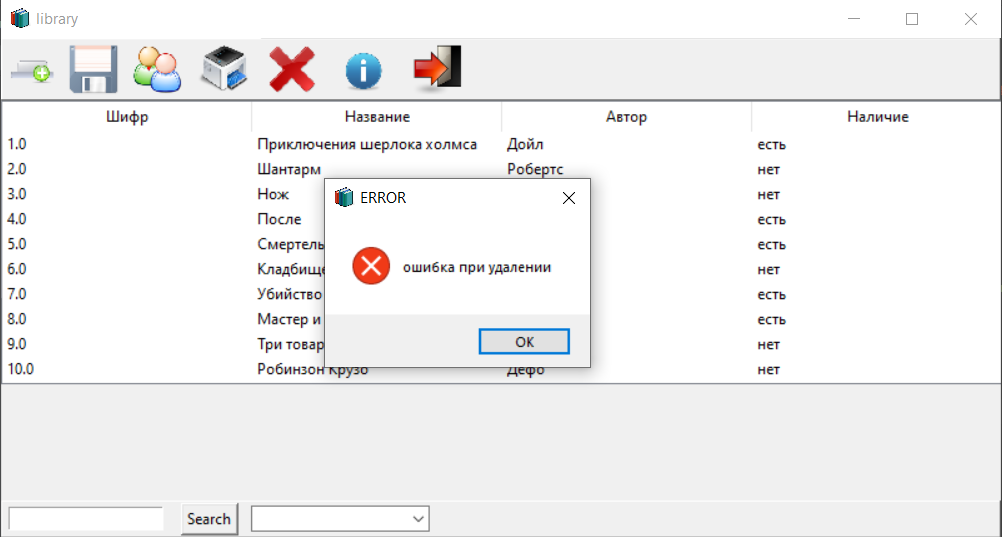
Макет экранной формы:



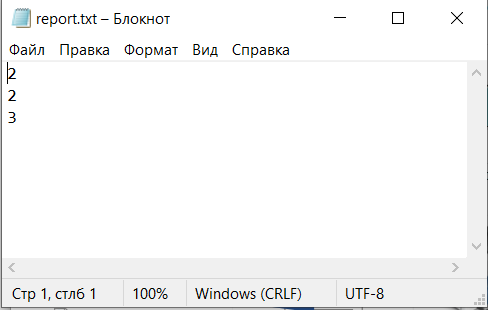
Демонстрация слушателя объекта new\_reader:



Обработка исключительных ситуаций:



Текстовый файл:



Текст программы:

Загрузка и сохранение в файл:

def change\_info(self, argument, count):  
 info\_file = open("../resources/report.txt", 'r')  
 txt = info\_file.read()  
 info\_mas = []  
 info\_mas.append(txt[0]) #count\_readers  
 info\_mas.append(txt[2]) #count\_new\_readers  
 info\_mas.append(txt[4]) #count\_new\_books  
 info\_mas[argument] = str(int(info\_mas[argument]) + count)  
 info\_file.close()  
 info\_file = open("../resources/report.txt", 'w')  
 info\_file.write(info\_mas[0] + '\n' + info\_mas[1] + '\n' + info\_mas[2] + '\0')  
 info\_file.close()

Обработка исключительных ситуаций:

def delete\_book\_func(self):  
 *"""removal book function"""* try:  
 a = self.table.selection()  
 res = messagebox.askquestion('Deleting', 'Вы хотите удалить книгу?')  
 if res:  
 self.table.delete(a)  
 self.table['height'] = self.table['height'] - 1  
 except:  
 mb = messagebox.showerror("ERROR", "ошибка при удалении")

Слушатели:

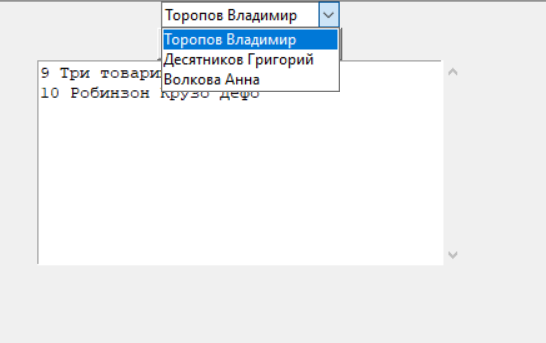
res = messagebox.showinfo('Добавление книги', 'Успешно')

# Лабораторная работа № 6

Цель: Знакомство с технологией обработки XML-документов и файлов

Проверка работоспособности:

*Создание XML:*

Исходная таблица:

Полученный XML:



Текст программы:

tree = ET.parse("../resources/readers.xml")  
self.root = tree.getroot()  
elements = []  
  
for reader in self.root:  
 elements.append(reader.attrib['name'])

text = ""  
current = self.root[0]  
current\_name = self.readers.get()  
  
for i in self.root:  
 if i.attrib['name'] == current\_name:  
 current = i  
 break  
for book in current:  
 text = text + book.attrib['code'] + ' ' + book[0].text + ' ' + book[1].text + '\n'  
self.books.insert(INSERT, text)

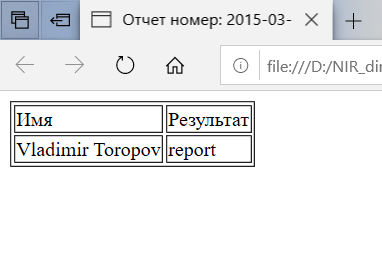
name = fio\_txt.get()  
reader = ET.SubElement(self.root, 'reader')  
reader.set('name', name)  
  
new\_xml = ET.tostring(self.root, 'UTF-8')  
xml\_file = open("../resources/readers.xml", 'wb')  
xml\_file.write(new\_xml)  
xml\_file.close()

# Лабораторная работа № 7 - 8

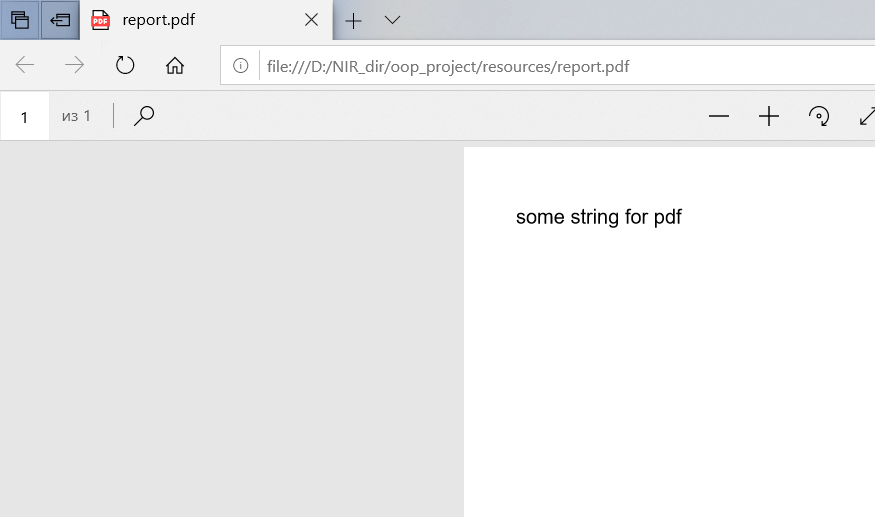
Цель: Знакомство со способами формирования PDF и HTML отчетов и правилами построения параллельных приложений в языке Python.

Проверка работоспособности:

HTML отчет:



*Возможно также сгенерировать PDF отчет (русские символы не отображаются)*



Текст программы:

Функция, отвечающая за работу потоков:

def threads\_word(self, wait\_thread, set\_thread):  
 wait\_thread.wait()  
 wait\_thread.clear()  
 txt\_file = open('../resources/threads.txt', 'a')  
 txt\_file.write("Поток")  
 txt\_file.close()  
 set\_thread.set()

t1 = threading.Event()  
t2 = threading.Event()  
  
self.first\_thread = threading.Thread(target=self.threads\_word, args=(t1, t2))  
self.second\_thread = threading.Thread(target=self.threads\_word, args=(t2, t1))  
  
self.first\_thread.start()  
self.second\_thread.start()  
  
t1.set()  
  
self.first\_thread.join()  
self.second\_thread.join()

Функция, отвечающая за генерацию отчетов:

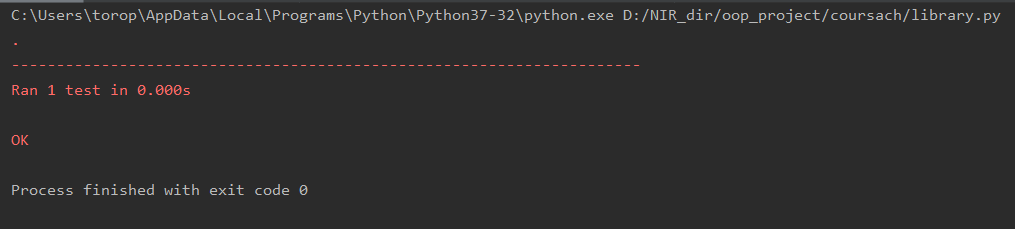
def save\_func(self):  
 *"""saving data function"""* txt = ""  
 rw = xlwt.Workbook('../resources/books\_write.xls')  
 sheet = rw.add\_sheet('Sheet1', cell\_overwrite\_ok=True)  
 for i in range(4):  
 sheet.write(0, i, self.table.heading(i)["text"])  
 for i in range(self.table["height"]):  
 value = ""  
 for j in range(4):  
 #print(self.table.item(i)["values"][j])  
 value = value + self.table.item(i)["values"][j] + ' '  
 sheet.write(i + 1, j, self.table.item(i)["values"][j])  
 txt = txt + value + '\n'  
 rw.save('../resources/books\_write.xls')  
  
 txt = "some string for pdf"  
 pdf = FPDF()  
 pdf.add\_page()  
 pdf.set\_font('Arial', size=12)  
 pdf.cell(200, 10, txt=txt)  
 pdf.output('../resources/report.pdf')  
  
 html\_template = """  
 <html>  
 <head>  
 <title>Отчет номер: {current\_date}</title>  
 </head>  
 <body>  
 <table border="1">  
 <tr>  
 <td>Имя</td>  
 <td>Результат</td>  
 </tr>  
 <tr>  
 <td>{arg\_w\_name}</td>  
 <td>{arg\_w\_res}</td>  
 </tr>  
 </table>  
 </body>  
 </html>  
 """  
 html\_file = open("../resources/report\_html.html", 'w')  
 html\_file.write(html\_template.format(current\_date="2015-03-31", arg\_w\_name="Vladimir Toropov", arg\_w\_res="report"))  
 mb = messagebox.showinfo('Success', 'Успешно сохранено')

# Лабораторная работа № 9

Цель: Знакомство с технологией модульного тестирования Python-приложения с использование системы Unittest.

Проверка работоспособности:

Все тесты выполнены успешно



Текст программы:

class Unittestclass(unittest.TestCase):  
 def test\_def(self):  
 self.assertEqual(5 + 5, 10)

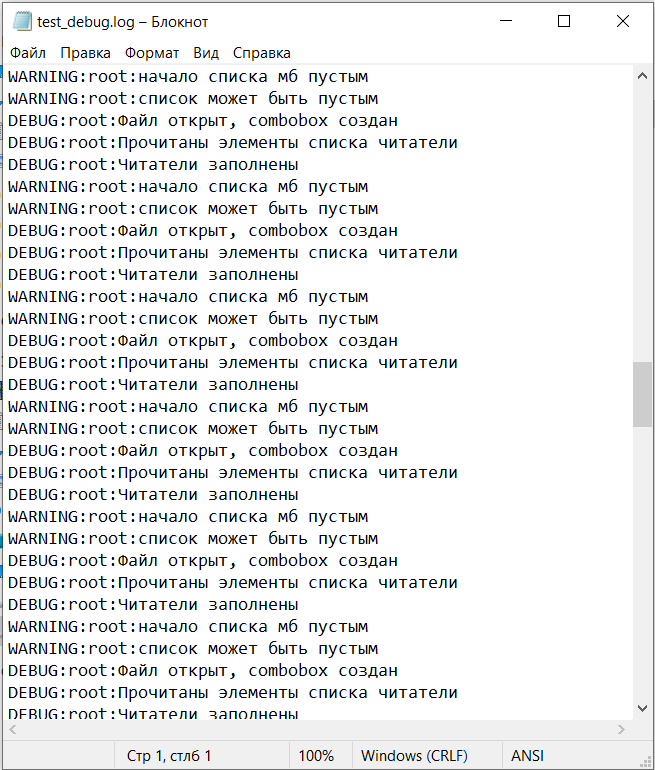
def print\_func(self):  
 *"""printing function"""* unittest.main()

# Лабораторная работа № 10

Цель: Знакомство c методами протоколирования работы приложения с использования библиотеки Logging.

Проверка работоспособности:

Пример протоколирования:



Пример кода:

logging.basicConfig(filename='test\_debug.log', level=logging.DEBUG)  
"""reading readers function"""  
tree = ET.parse("../resources/readers.xml")  
self.root = tree.getroot()  
self.readers = Combobox(self, state='readonly')  
elements = []  
logging.debug("Файл открыт, combobox создан")  
  
for reader in self.root:  
 elements.append(reader.attrib['name'])  
  
logging.debug("Прочитаны элементы списка читатели")