

## **Практико-ориентированный подход к обучению языка программирования Python в системе дополнительного образования**

Березин И.С.

*vasbete@gmail.com*

*Саратовский государственный университет имени Н.Г. Чернышевского*

В работе рассматривается один из вариантов комплекса практико-ориентированных лабораторных работ, которые помогут детям лучше усвоить материал.

**Ключевые слова:** языки программирования, python, практико-ориентированный подход, дополнительное образование.

Актуальность разработки практико-ориентированного обучения заключается в том, что данный подход позволяет значительно повысить эффективность обучения. Этому способствует система отбора содержания учебного материала, помогающая учащимся оценивать значимость, практическую востребованность приобретаемых знаний и умений. В процессе обучения широко используются творческие домашние задания, учащиеся

получают возможность обращаться к своей фантазии, к творчеству. В практико-ориентированном учебном процессе не только применяется имеющийся у учащихся жизненный опыт, но и формируется новый опыт на основе вновь приобретаемых знаний. Данный опыт становится основой развития учащихся.

Ключевая задача практико-ориентированного обучения заключается в создании условий обучения на основе единства логического и эмоционально-образного, а также социального компонентов содержания; формировании практического опыта при решении задач, практико-ориентированных проектов и применение такого опыта в профессиональной сфере; раскрытие творческого, социального и эмоционального потенциала учащегося [1].

В качестве примера такого практико-ориентированного обучения, давайте рассмотрим, комплекс лабораторных работ по созданию чат-бота Вконтакте на языке Python.

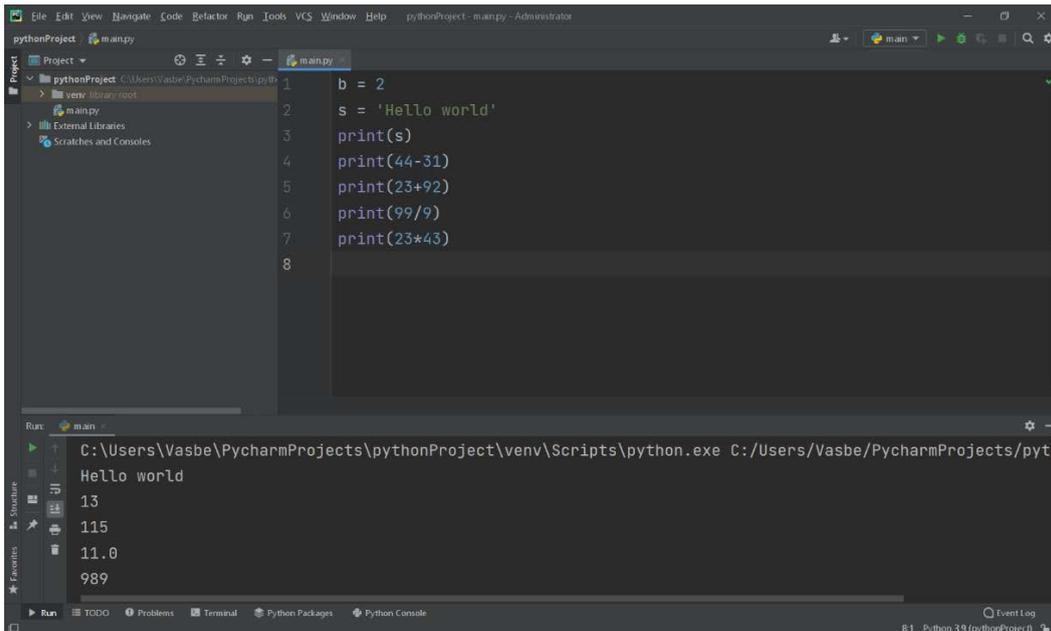
В комплексе представлены 6 лабораторных работ, которые помогут ученикам ознакомиться с основными разделами программирования на языке Python и создать своего собственного чат-бота Вконтакте. Тема чат-бот Вконтакте была выбрана не случайно, так как многие ученики заинтересованы в создании современного инструмента автоматизированного общения с пользователем, созданием простого искусственного интеллекта.

Комплекс лабораторных работ можно разделить на 2 части. В первых трёх лабораторных работах описываются базовые темы, которые потребуются для создания чат-бота, это темы: «Установка и настройка среды программирования», где ученики скачивают и устанавливают IDE для разработки и базовое программное обеспечение для работы с Python [2]. «Работа с типами данных» (рис. 1), в данной теме мы рассматриваем для чего нужны типы данных, какие они могут быть, что можно делать с различными типами данных. «Ветвление», тема ветвление необходима для создания логики бота, на данных уроках мы рассматриваем, как правильно описать базовые логические высказывания, знакомимся с синтаксисом и командами. «Функции», раздел функции нужен для создания чистого кода, данный раздел поможет ученикам в будущем не писать одинаковый код несколько раз в одной программе, на уроке рассматриваются команды функций и их использование. Отдельное внимание уделяется разделу «Строки» (рис. 2), так как созданный бот должен будет выполнять роль школьного помощника, который будет общаться с пользователем с помощью строк, подскажет расписание, какие домашние задания нужно будет подготовить на завтра, подскажет, где можно будет провести свой досуг. [4]

В остальных трёх лабораторных работах описывается процесс создания самого чат-бота и наполнение его контентом. Первая тема знакомит учеников с библиотеками `vk_api`, помогает настроить бота для дальнейшей работы с ним и создаётся простой пример распознавания текста ботом. Во второй лабораторной работе, ученики создают функцию выдачи расписания уроков по специальному сообщению (рис. 3). Третья тема направлена на самостоятельное создание функции мест для проведения досуга. [2]

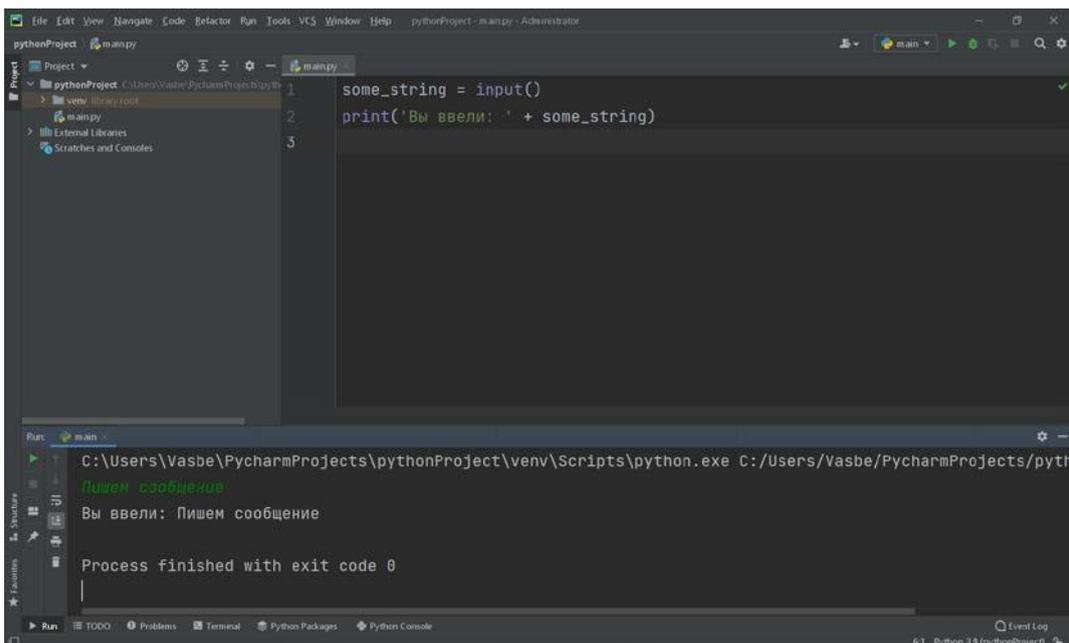
Предлагаю подробнее рассмотреть первое занятие из второй части лабораторных работ, где ученики знакомятся с библиотекой `vk_api`.

После ознакомления с основными возможностями языка Python, можно приступить к созданию чат-бота. Но перед этим рассмотрим еще одно понятие – библиотека. Библиотека – это написанный код, который позволяет программистам взаимодействовать с теми или иными сервисами и функциями. Библиотеки ранее были написаны другими программистами, для простоты работы, вместо того, чтобы писать огромные массивы кода, мы будем пользоваться всего парой строчек кода, написанного вместо нас ранее.



```
pythonProject - main.py - Administrator
pythonProject C:\Users\Vasbe\PycharmProjects\pythonProject
venv library root
main.py
External Libraries
Scratches and Consoles
1 b = 2
2 s = 'Hello world'
3 print(s)
4 print(44-31)
5 print(23+92)
6 print(99/9)
7 print(23*43)
8
Run main
C:\Users\Vasbe\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Vasbe/PycharmProjects/pyth
Hello world
13
115
11.0
989
Run TODO Problems Terminal Python Packages Python Console
Event Log
6.1 Python 3.9 (pythonProject)
```

Рис. 1. Выполненное задание на тему типы данных.



```
pythonProject - main.py - Administrator
pythonProject C:\Users\Vasbe\PycharmProjects\pythonProject
venv library root
main.py
External Libraries
Scratches and Consoles
1 some_string = input()
2 print('Вы ввели: ' + some_string)
3
Run main
C:\Users\Vasbe\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Vasbe/PycharmProjects/pyth
Пишем сообщение
Вы ввели: Пишем сообщение
Process finished with exit code 0
Run TODO Problems Terminal Python Packages Python Console
Event Log
6.1 Python 3.9 (pythonProject)
```

Рис. 2. Первое задание на тему строки.

Для того чтобы подключить библиотеку для бота в VK нужно написать в начале файла команду подключения библиотеки:

```
import vk_api
```

Однако, если запустить эту программу, мы получим ошибку (рис. 4) [2]

Эта ошибка возникает, поскольку у нас отсутствует данная библиотека, и её необходимо загрузить. Заходим во вкладку «Terminal» и вводим команду (рис. 5). [1] Для выполнения этой команды необходимо нажать клавишу «Enter».

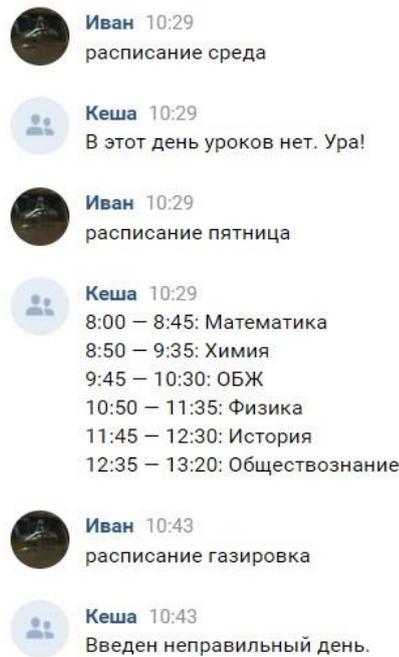


Рис. 3. Итог по окончании урока

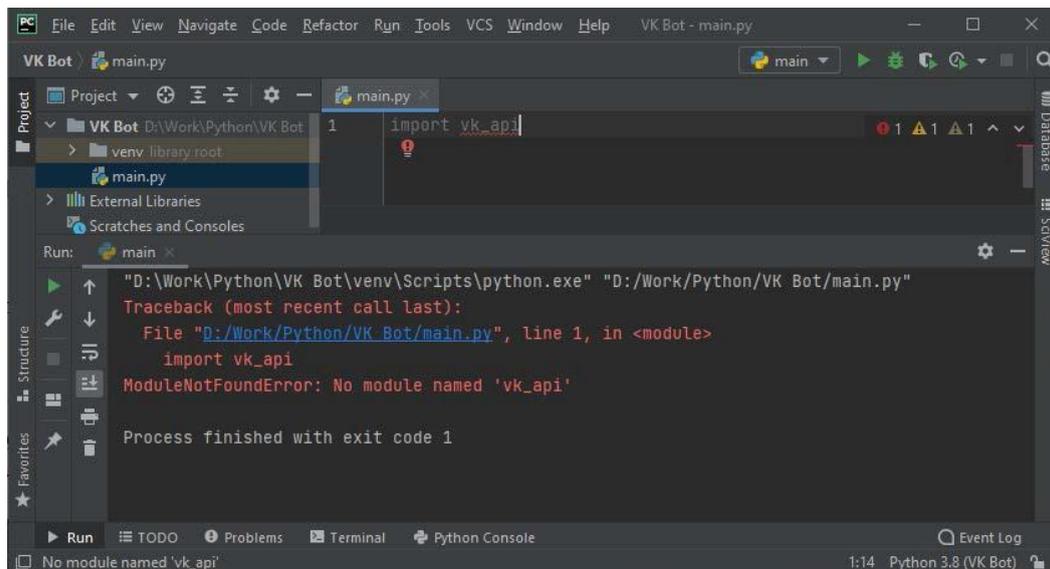


Рис. 4. Ошибка

По завершении загрузки библиотеки мы сможем использовать команды из данной библиотеки, но нам понадобятся не все функции. Для удобства

написания кода мы получим из библиотеки лишь необходимые функции. Для этого напишем строку следующей строку:

```
from vk_api.longpoll import VkLongPoll, VkEventType
```

После импортирования в программу необходимых библиотек и функций, можно приступить к написанию кода. [1]

В первую очередь сохраним в переменную token строку, содержащую ключ доступа к API, полученный в лабораторной работе 1:

```
token = "<ключ доступа к API>"
```

Для подключения к VK напишем следующую строку:

```
vk = vk_api.VkApi(token)
```

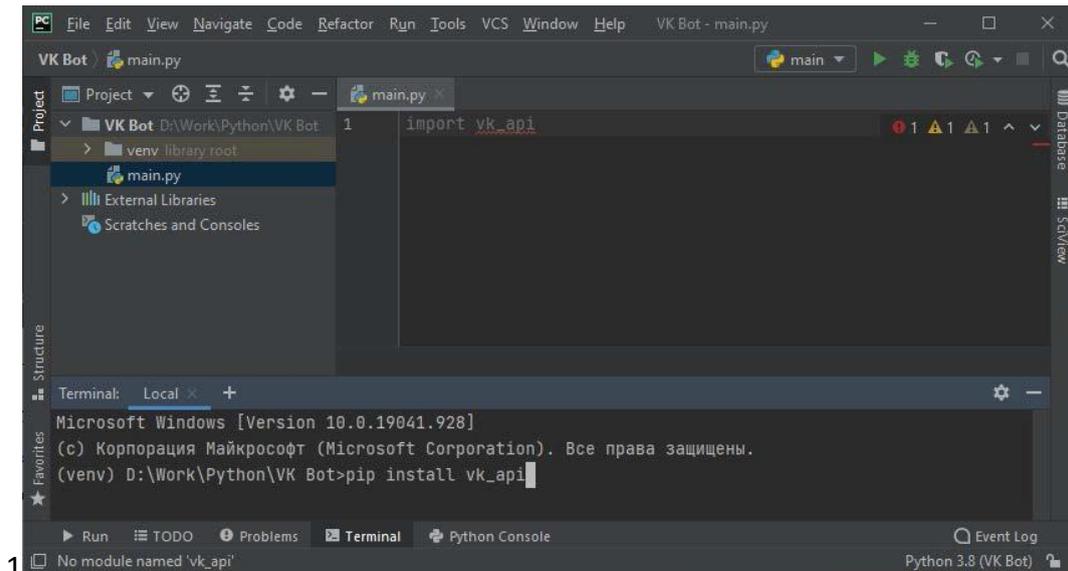


Рис. 5. Команда

В этой строке мы вызываем функцию `VkApi` из библиотеки `vk_api`, передавая ей полученный ранее ключ доступа к API. В качестве результата эта функция вернет объект, с помощью которого мы сможем выполнять различные действия с VK (если будет передан корректный ключ API). Этот объект сохраняется в переменную `vk`.

Для работы чат-бота нам необходимо постоянно проверять, какие события с ним происходят, будь то отправка ему сообщения или наш вход на сайт. Для этого нам понадобится еще один объект, который мы создадим следующей строкой:

```
longpoll = VkLongPoll(vk)
```

Этот объект позволит нам обрабатывать происходящие с ботом события.

Таким образом, на данном этапе код программы должен иметь следующий

вид:

```
import vk_api
from vk_api.longpoll import VkLongPoll, VkEventType
import random
token = "<ключ доступа>"
vk = vk_api.VkApi(token=token)
longpoll = VkLongPoll(vk)
```

Перейдем к написанию основного кода программы. Для этого создадим функцию `main()`, в которой напишем следующий код:

```
def main():
    for event in longpoll.listen():
        if event.type == VkEventType.MESSAGE_NEW:
            if event.to_me:
                request = event.text.lower()
```

Разберем написанный код. Внутри функции `main` при помощи цикла `for..in` мы перебираем все события из списка происходящих с ботом событий, который возвращается при помощи функции `longpoll.listen()` [3]. Внутри этого цикла мы проверяем при помощи оператора `if` тип этого события. Если тип события – новое сообщение (`VkEventType.MESSAGE_NEW`), то дополнительно проверим, кому именно пришло новое сообщение – нам от бота или боту от нас. Если сообщение пришло именно от нас к боту (`event.to_me`), то преобразуем текст сообщения в нижний регистр и сохраним его в переменную `request`.

Теперь после того, как мы получили текст отправленного боту сообщения, можно описывать логику отправки ему ответов. Для удобства написания кода создадим перед функцией `main` функцию `write_message`, которая будет заниматься отправкой боту сообщения.[1]

Код функции `write_message`:

```
def write_message(user_id, message):
    vk.method('messages.send', {
        'user_id': user_id,
        'message': message,
        'random_id': random.randint(0, 100000)
    })
```

Эта функция в качестве параметров получает идентификатор пользователя, которому мы хотим отправить сообщение, и текст самого сообщения. Внутри неё мы вызываем метод ВК «Отправить сообщение» (`messages.send`), передав ему в качестве параметров искомые ID пользователя и текст сообщения, а также уникальный идентификатор сообщения, который мы получим, сгенерировав случайное число от 0 до 100000. Для работы функции `randint` в начале программы необходимо добавить следующую строку:

```
import random
```

Используем полученную функцию `write_message` для отправки сообщения. Изменим код функции `main` следующим образом:

```
if event.to_me:
    request = event.text.lower()
    if request == "привет:":
        write_message(event.user_id, 'И тебе привет, друг мой')
    else:
        write_message(event.user_id, 'Извини, не могу понять, что ты мне написал')
```

После получения сообщения в нижнем регистре, проверим, равно ли оно слову «привет». Если это так, отправим ответным сообщением приветствие пользователю. В противном случае отправим сообщение о том, что бот не может понять, что мы ему написали.

Проверим работоспособность полученного бота. Для этого запустим программу и попробуем написать в сообщения группы VK некоторое сообщение. Результаты проверки работы бота (рис. 6).

Данные лабораторные работы направлены на увлекательное изучение базовых основ языка программирования python, так как тема по созданию чат-ботов интересует многих учеников 7-11 классов, так же данный курс позволит осветить некоторые внутренние процессы работы в IT сфере, таким образом данный курс может привлечь некоторых учеников старшей и средней школы к IT специализации.

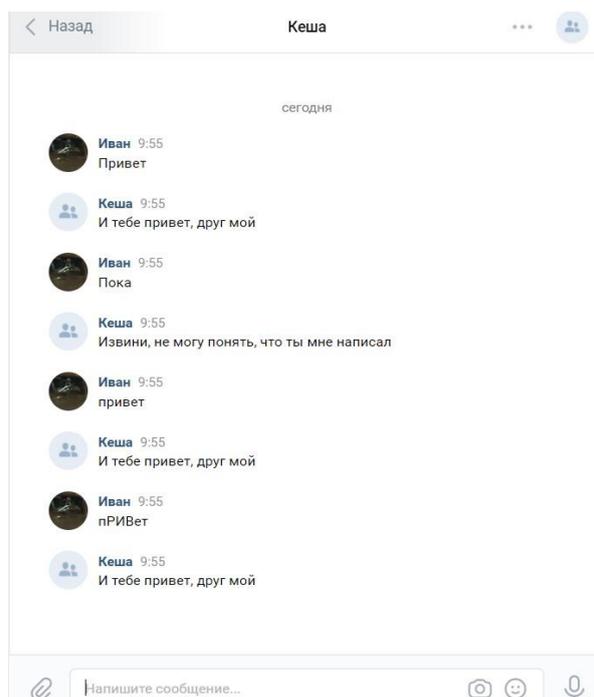


Рис. 6. Итог работы программы

Подводя итог, мы пришли к выводу, что практико-ориентированные лабораторные работы в дополнительном образовании IT сферы могут не только повысить уровень компьютерной грамотности, но и привлечь молодое поколение к IT специальностям.

### Список литературы

- [1] Багаев, А. В. Возможности решения типовых задач в базовом курсе информатики и ИКТ на языке программирования Python / А. В. Багаев, Н. А. Александрова // Информационные технологии в образовании : Материалы VI Всероссийской научно-практической конференции, Саратов, 06–07 ноября 2014 года / Саратовский государственный университет имени Н.Г. Чернышевского. – Саратов: ООО "Издательский центр "Наука", 2014. – С. 4-6.
- [2] Сайт документации vk\_api: <https://vk.com/dev/manuals>
- [3] Сайт python: <https://www.python.org/>
- [4] Python Для детей. Самоучитель по программированию / Джейсон Бриггс; пер. с англ. Станислава Ломакина; [науч. Ред. Д. Абрамова]. – М. : Манн, Иванов и Фербер, 2017. – 320с.