

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

Дисциплина: «языки программирования»

ОТЧЕТ

по лабораторной работе №7

Выполнил: студент 2 курса группы ИТС-21-1

Снадный Михаил Сергеевич

Проверил: доцент кафедры инфокоммуникаций

Воронкин Роман Александрович

Работа защищена с оценкой: _____

Ставрополь, 2022

Тема:

Разработка приложений с интерфейсом командной строки (CLI) в Python3

Цель работы:

приобретение построения приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

- 1) Создадим общедоступный репозиторий на GitHub (<https://github.com/peach909/17.git>)
- 2) Решим задачи с помощью языка программирования Python3. И отправим их на GitHub.

Проработал пример 1

File Edit Format Run Options Window Help

```
action="store",
required=True,
help="The worker's name"
)
add.add_argument(
    "-p",
    "--post",
    action="store",
    help="The worker's post"
)
add.add_argument(
    "-y",
    "--year",
    action="store",
    type=int,
    required=True,
    help="The year of hiring"
)
_ = subparsers.add_parser(
    "display",
    parents=[file_parser],
    help="Display all workers"
    Результаты работы программы:
)
select = subparsers.add_parser(
    "select",
    parents=[file_parser],
    help="Select the workers"
)
select.add_argument(
    "-p",
    "--period",
    action="store",
    type=int,
    required=True,
    help="The required period"
)
args = parser.parse_args(command_line)
is_dirty = False
if os.path.exists(args.filename):
    workers = load_workers(args.filename)
else:
    workers = []
if args.command == "add":
    workers = add_worker(
        workers,
        args.name,
        args.post,
        args.year
    )
is_dirty = True
elif args.command == "display":
    display_workers(workers)
elif args.command == "select":
    selected = select_workers(workers, args.period)
    display_workers(selected)
if is_dirty:
    save_workers(args.filename, workers)
if __name__ == "__main__":
    main()
```

```
# cli.py
import click

@click.command()
def main():
    print("I'm a beautiful CLI 🌟")

if __name__ == "__main__":
    main()
```

Рисунок 1. Индивидуальное задание

1. В чем отличие терминала и консоли?

Консоль - это совокупность устройств, которые позволяют вам взаимодействовать с устройством. Раньше консолью мог быть обычный принтер (в принципе и сейчас может), тогда это будет текстовая консоль/терминал. Не надо путать с текстовым интерфейсом. Терминал - По сути это так же устройство для ввода и вывода информации но>. Консоль, это уже обёртка над терминалом.

2. Что такое консольное приложение?

Консольным приложением называется программа, которая не имеет графического интерфейса - окон, и которая работает в текстовом режиме в черно-белой консоли

3. Какие существуют средства языка программирования Python для построения приложений командной строки?

Click — это Python-пакет для создания красивых интерфейсов командной строки с минимальным количеством кода и возможностью компоновки. Это один из лучших Python-пакетов для создания CLI, и с ним очень удобно работать.

4. Какие особенности построение CLI с использованием модуля sys ?

Модуль sys реализует аргументы командной строки в простой структуре списка с именем sys.argv На практике для правильной обработки входных

данных требуется модуль `sys`. Для этого необходимо предварительно загрузить как модуль `sys`, так и модуль `getopt`. ... Генерация интерфейсов командной строки (CLI) с помощью Fire в Python. Модуль `Shutil` в Python.

5. Какие особенности построение CLI с использованием модуля `getopt`?

чтобы упростить написание кода, придерживающегося стандартных соглашений. Функция GNU `getopt_long()`, является совместимой с `getopt()`, а также упрощает разбор длинных опций. `getopt`. Объявление: `#include <unistd.h> int getopt(int argc, char *argv[], const char *optstring);`

6. Какие особенности построение CLI с использованием модуля `argparse`?

Использование модуля `argparse` в Python для создания интерфейса командной строки, обработки позиционных и необязательных аргументов, их комбинирование с подробными примерами. ... Интерфейс командной строки в Python также известен как CLI,

Вывод: приобрели навыки построения приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x.