

*Государственное образовательное учреждение высшего
профессионального образования*

**«Московский государственный технический
университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**

**ФАКУЛЬТЕТ Информатика и системы управления
КАФЕДРА Системы обработки информации и управления**

**Отчёт по лабораторной работе
по курсу
«Разработка интернет-приложений»
Введение в Python**

Исполнитель:
студентка группы **РТ5-51**
Карасева А. Д.
Преподаватель: **Гапанюк Ю.Е.**

Москва, 2017

Цель работы: ознакомиться с базовым синтаксисом Python. Дополнительная задача – знакомство с Git и GitHub.

Содержание файла «arr_algs.py»:

```
def find_max(array):
    max_value = 0
    if len(array) == 0:
        return -1
    for v in array:
        if v >= max_value:
            max_value = v
    return max_value

def average(array):
    if len(array) == 0:
        return -1
    sum = 0
    for m in array:
        sum += m
    average_value = sum / len(array)
    return average_value

def main():
    mas1 = [1, 2, 3]

    print ("max =" + str(find_max(mas1)))

    print ("average=" + str(average(mas1)))
main()
```

Функция «max (array)» реализует нахождение минимума в массиве. Нахождение среднего арифметического в массиве осуществляет функция «average (array)».

Результат работы программы:

max =3

average=2.0

Содержание файла «str_algs.py»:

```
def perevorot_stroki(stroka_do):
    index = len(stroka_do) - 1
    stroka_posle = ''
```

```

while index >= 0:
    stroka_posle += stroka_do[index]
    index-=1
return stroka_posle

```

```

def main():
    stroka="kukushka"
    print ("stroka:" + stroka)
    print ("perevorot stroki:" + str(perevorot_stroki(stroka)))
main()

```

Функция «perevorot_stroki(stroka_do)» выполняет переворот строки.

Результат работы программы:

stroka:kukushka

perevorot stroki: akhsukuk

Содержание файла «dict_algs.py»:

```

def emps_with_children_elder_than_(emps, children_age):
    filtered_emps = []
    for emp in emps:
        if 'children' in emp:
            for ch in emp['children']:
                if ch['age'] > children_age:
                    filtered_emps.append(emp['name'])
                    break
    return filtered_emps

```

```

ivan = {
    'name': 'ivan',
    'age': 34,
    'children': [{
        'name': 'vasja',
        'age': 12,
    }, {
        'name': 'petia',
        'age': 10,
    }]
}

```

```

darja = {

```

```
'name': 'darja',  
'age': 41,  
'children': [{  
    'name': 'kirill',  
    'age': 21,  
}, {  
    'name': 'pavel',  
    'age': 15,  
}]  
}
```

```
all_emps = [ivan, darja]
```

```
def main():  
    print('emps with children elder 18:')  
    print(emps_with_children_elder_than_(all_emps, 18))  
main()
```

Программа выводит имена сотрудников, у которых есть дети старше 18 лет.
Результат работы программы:
['darja']