

**Тема:** Составление программ со словарями в IDE PyCharm Community.

**Цель:** Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со словарями в IDE PyCharm Community.

**Постановка задачи №1.**

Выполнить сортировку словаря `d = {'a': 1, 'b': 2, 'c': 3}`

**Тип алгоритма:** Циклический

**Текст программы:**

```
S = str(input("Введите значение для строки S: "))
d = {'a': 1, 'b': 2, 'c': 3}

reversedDict = {key: value for key, value in reversed(d.items())}
print(reversedDict)
```

**Протокол работы программы:**

`{'c': 3, 'b': 2, 'a': 1}`

Process finished with exit code 0

**Постановка задачи №2.**

Дана строка `'2020год -16 -10 -6 4 20 32 36 32 32 15 1 -15'`, отражающая средние температуры по месяцам в году. Преобразовать информацию из строки в словарь, с использованием функции `find` найти среднюю и минимальные температуры, результаты вывести на экран.

**Тип алгоритма:** Циклический

### Текст программы:

```
"""
Дана строка '2020год -16 -10 -6 4 20 32 36 32 32 15 1 -15', отражающая
средние температуры по месяцам в году.
Преобразовать информацию из строки в словарь, с использованием функции найти
среднюю и минимальные температуры,
результаты вывести на экран.
"""
import json

def str_with_mounths_to_dict(_stroke: str, _mounths: list) -> dict:
    """Преобразование строки в словарь"""
    local_calendar = {}
    for i in range(len(_stroke.split(" "))):
        local_calendar[_mounths[i]] = _stroke.split(" ")[i]
    return local_calendar

def int_list_of_temperatures(calendar: dict) -> list:
    """Кладем все температуры в список для дальнейшего использования"""
    temperatures = list(calendar.values())[1:]
    return list(map(int, temperatures))

def average_temperature() -> int:
    """Находим среднюю температуру"""
    temperatures = int_list_of_temperatures(dict_calendar)
    return round(sum(temperatures) / len(temperatures))

def min_temperature() -> int:
    """Находим минимальную температуру"""
    temperatures = int_list_of_temperatures(dict_calendar)
    return min(temperatures)

data = [
    "Year", "January", "February", "March", "April", "May", "June",
    "July", "August", "September", "October", "November", "December"
]
S = "2020год -16 -10 -6 4 20 32 36 32 32 15 1 -15"
dict_calendar = str_with_mounths_to_dict(S, data)

dict_calendar["average_temperature"] = average_temperature()
dict_calendar["min_temperature"] = min_temperature()

# тут ужасно неудобный вывод для просмотра
print(dict_calendar)

# поэтому добавляю в json`чик, чтобы удобнее было просматривать
with open("calendar.json", "w", encoding="utf-8") as file:
    json.dump(dict_calendar, file, ensure_ascii=False, indent=4)
```

### Протокол работы программы:

```
{'Year': '2020год', 'January': '-16', 'February': '-10', 'March': '-6', 'April': '4', 'May': '20', 'June':
'32', 'July': '36', 'August': '32', 'September': '32', 'October': '15', 'November': '1', 'December': '-
15', 'average_temperature': 10, 'min_temperature': -16}
```

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия №7 выработал навыки

составления программ строками в IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные кода выложены на GitHub.