

Упражнение 1: Операции со строкой

Решение:

```
string = "Python Programming"
print(f"Длина строки: {len(string)}")
print(f"Символ по индексу 7: {string[7]}")
print(f"Последние 3 символа: {string[-3:]}")
if "gram" in string:
    print("Подстрока 'gram' содержится в строке.")
else:
    print("Подстрока 'gram' не содержится в строке.")
```

Упражнение 2: Методы строк и форматинг

Решение:

```
email = " USER@DOMAIN.COM "
email = email.strip().lower()
print(f"Отформатированный email: {email}")
username, domain = email.split('@')
print(f"Имя пользователя: {username}, Домен: {domain}")
formatted_string = f"Username: {username}, Domain: {domain}"
print(formatted_string)
```

Упражнение 3: Методы списка

Решение:

```
fruits = ["apple", "banana"]
fruits.append("orange")
print(f"После добавления 'orange': {fruits}")
fruits.insert(1, "grape")
print(f"После вставки 'grape': {fruits}")
fruits.remove("banana")
print(f"После удаления 'banana': {fruits}")
fruits.sort()
print(f"После сортировки: {fruits}")
fruits.reverse()
print(f"После переворота: {fruits}")
```

Упражнение 4: List comprehension и словари

Решение:

```
words = ["hello", "world", "python", "code"]
word_lengths = [len(word) for word in words]
print(f"Список длин слов: {word_lengths}")
long_words = [word for word in words if len(word) > 4]
print(f"Список слов длиннее 4 символов: {long_words}")
word_length_dict = {word: len(word) for word in words}
print(f"Словарь слово:длина: {word_length_dict}")
```

Упражнения 5 (Опционально):

Решение:

```
def find_two_sum(nums, target):
    num_map = {}
    for index, num in enumerate(nums):
        complement = target - num
        if complement in num_map:
            return [num_map[complement], index]
        num_map[num] = index
    return None
```

Пример 1

```
nums1 = [2, 7, 11, 15]
target1 = 9
result1 = find_two_sum(nums1, target1)
print(f"Пример 1: nums = {nums1}, target = {target1}, результат = {result1}")
```

Пример 2

```
nums2 = [3, 2, 4]
target2 = 6
result2 = find_two_sum(nums2, target2)
print(f"Пример 2: nums = {nums2}, target = {target2}, результат = {result2}")
```

Часть 2: Функции и ООП

Упражнение 1: Функции без параметров

Решение:

```
import datetime
def show_current_time():
    now = datetime.datetime.now()
    print("Текущая дата и время:", now)
show_current_time()
```

Упражнение 2: Функции с параметрами

Решение:

```
def add_vat(price, nds=0.20):

    return price + price * nds

prices = [1000, 3499, 250]

for price in prices:
    final_price = add_vat(price)
    print(final_price)
```

Упражнение 3 (Опционально)

```
def calculate_average_score(scores, ignore_lowest=False):
    if not scores:
        return None

    if ignore_lowest and len(scores) > 1:
        scores = sorted(scores)[1:]

    return sum(scores) / len(scores)

student_data = [
    {'name': 'Алексей', 'scores': [85, 92, 78, 95]},
    {'name': 'Марина', 'scores': [65, 70, 58, 82]},
    {'name': 'Светлана', 'scores': [98, 95, 100]}
]
print("Средние баллы со всеми оценками:")
for student in student_data:
    average_score = calculate_average_score(student['scores'])
    print(f"{student['name']}: {average_score:.2f}")

print("\nСредние баллы с отброшенной худшей оценкой:")
for student in student_data:
```

```
average_score = calculate_average_score(student['scores'], ignore_lowest=True)
print(f"{student['name']}: {average_score:.2f}")
```