

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

**«Элементы объектно-ориентированного программирования в языке
Python»**

ОТЧЕТ
по лабораторной работе №4.1
дисциплины
«Основы программной инженерии»

Выполнила:

Кувшин Ирина Анатольевна
2 курс, группа ПИЖ-б-о-21-1,
011.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

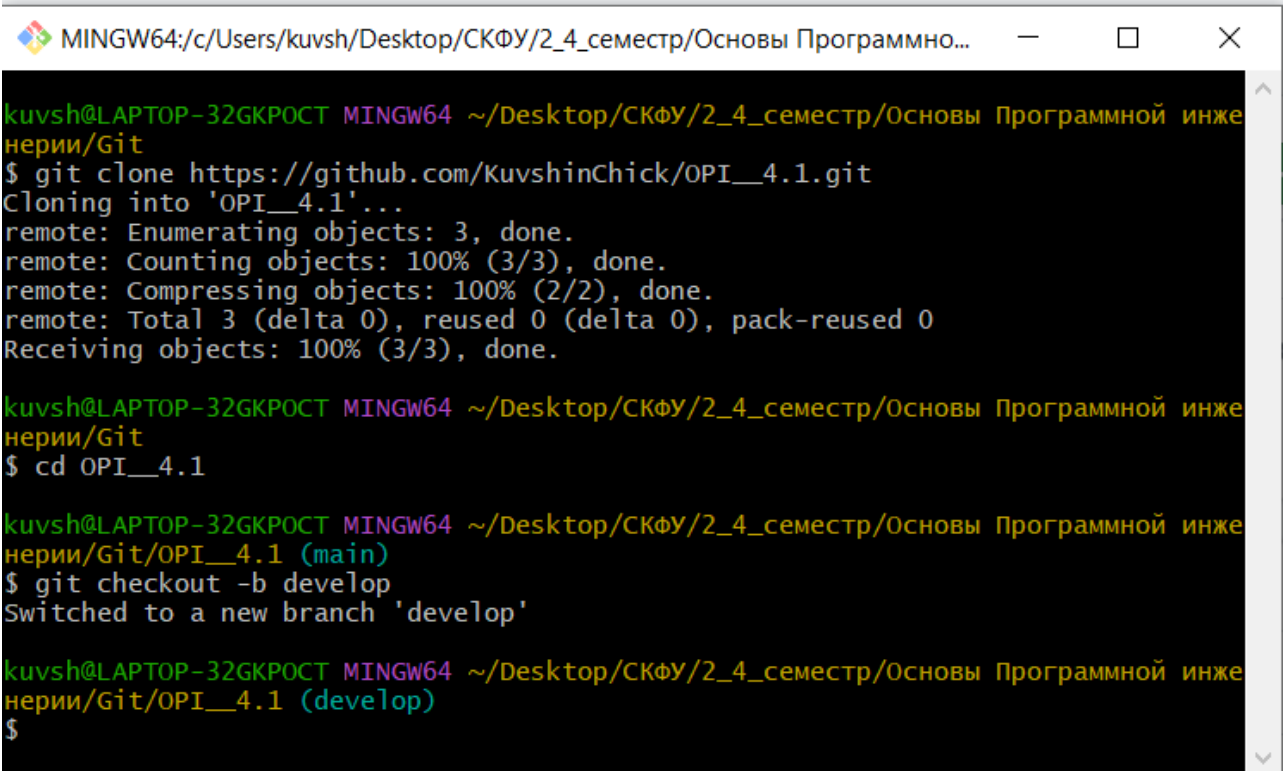
Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Цель работы: приобретение навыков по работе с классами и объектами при написании программ с помощью языка программирования Python версии 3.x.

Ход работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.
6. Создайте проект PyCharm в папке репозитория.
7. Проработайте примеры лабораторной работы.



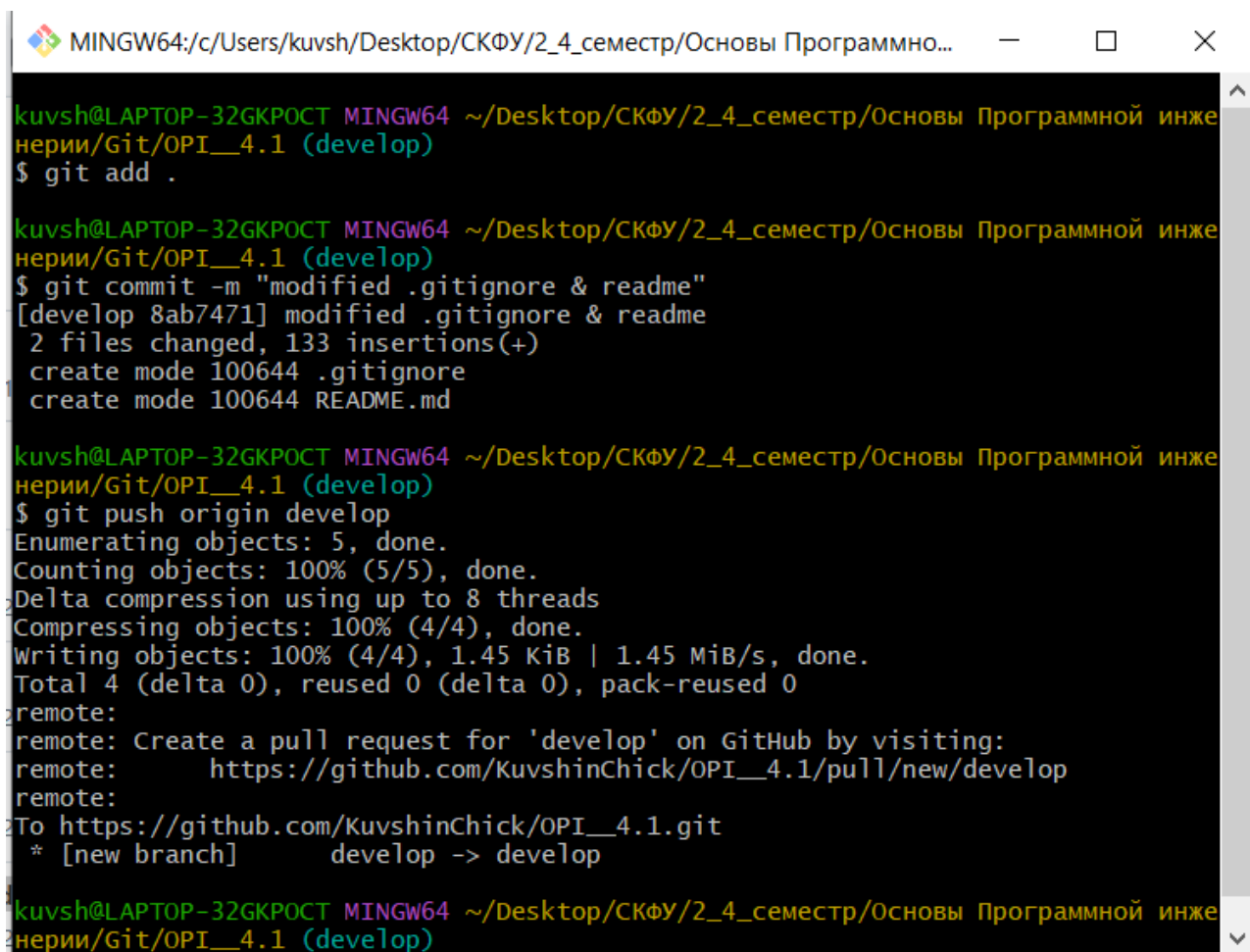
```
MINGW64:/c/Users/kuvsh/Desktop/СКФУ/2_4_семестр/Основы Программно...
kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программно...
$ git clone https://github.com/KuvshinChick/OPI__4.1.git
Cloning into 'OPI__4.1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программно...
$ cd OPI__4.1

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программно...
$ git checkout -b develop
Switched to a new branch 'develop'

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программно...
$
```

Рисунок 4.1.1 – Клонирование репозитория и создание ветки develop



```
kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программной инженерии/Git/OPI__4.1 (develop)
$ git add .

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программной инженерии/Git/OPI__4.1 (develop)
$ git commit -m "modified .gitignore & readme"
[develop 8ab7471] modified .gitignore & readme
2 files changed, 133 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программной инженерии/Git/OPI__4.1 (develop)
$ git push origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.45 KiB | 1.45 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/KuvshinChick/OPI__4.1/pull/new/develop
remote:
To https://github.com/KuvshinChick/OPI__4.1.git
 * [new branch]      develop -> develop

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программной инженерии/Git/OPI__4.1 (develop)
```

Рисунок 4.1.2 – Обновление .gitignore и readme

8. Выполните индивидуальные задания. Приведите в отчете скриншоты работы программ решения индивидуального задания.

Задание 1

Парой называется класс с двумя полями, которые обычно имеют имена `first` и `second`. Требуется реализовать тип данных с помощью такого класса. Во всех заданиях обязательно должны присутствовать:

метод инициализации `__init__` ;

метод должен контролировать значения аргументов на корректность;

ввод с клавиатуры `read` ;

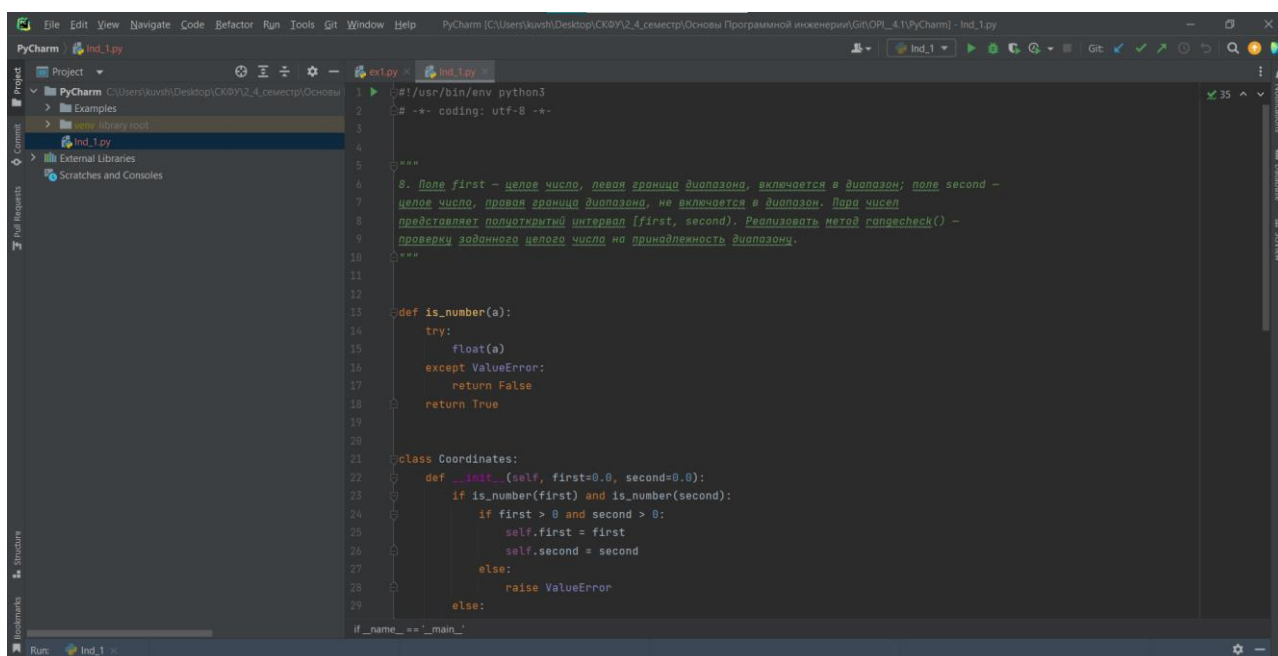
вывод на экран `display` .

Реализовать внешнюю функцию с именем `make_тип()` , где `тип` — тип реализуемой структуры. Функция должна получать в качестве аргументов

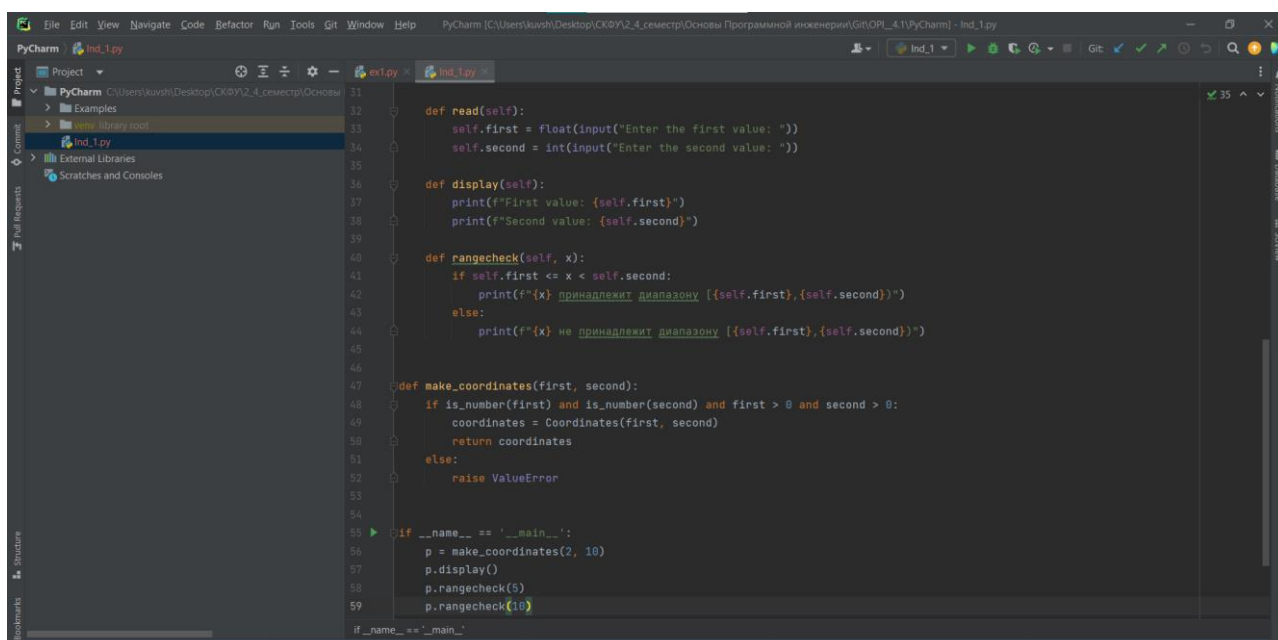
значения для полей структуры и возвращать структуру требуемого типа. При передаче ошибочных параметров следует выводить сообщение и заканчивать работу.

Номер варианта необходимо уточнить у преподавателя. В раздел программы, начинающийся после инструкции `if __name__ == '__main__':` добавить код, демонстрирующий возможности разработанного класса.

8. Поле `first` — целое число, левая граница диапазона, включается в диапазон; поле `second` — целое число, правая граница диапазона, не включается в диапазон. Пара чисел представляет полуоткрытый интервал `[first, second)`. Реализовать метод `rangecheck()` — проверку заданного целого числа на принадлежность диапазону.

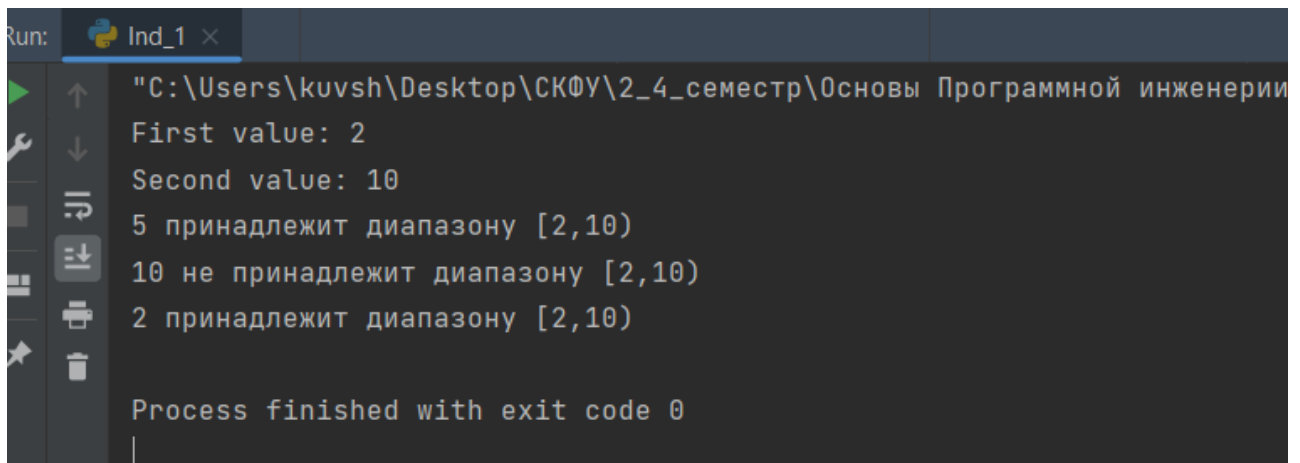


```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5
6 """
7 8. Поле first – целое число, левая граница диапазона, включается в диапазон; поле second –
8 целое число, правая граница диапазона, не включается в диапазон. Пара чисел
9 представляет полуоткрытый интервал [first, second). Реализовать метод rangecheck() –
10 проверку заданного целого числа на принадлежность диапазону.
11 """
12
13
14 def is_number(a):
15     try:
16         float(a)
17     except ValueError:
18         return False
19     return True
20
21
22 class Coordinates:
23     def __init__(self, first=0.0, second=0.0):
24         if is_number(first) and is_number(second):
25             if first > 0 and second > 0:
26                 self.first = first
27                 self.second = second
28             else:
29                 raise ValueError
30         else:
31             raise ValueError
32
33 if __name__ == '__main__':
```



```
31
32
33 def read(self):
34     self.first = float(input("Enter the first value: "))
35     self.second = int(input("Enter the second value: "))
36
37 def display(self):
38     print(f"First value: {self.first}")
39     print(f"Second value: {self.second}")
40
41 def rangecheck(self, x):
42     if self.first <= x < self.second:
43         print(f"{x} принадлежит диапазону [{self.first}, {self.second})")
44     else:
45         print(f"{x} не принадлежит диапазону [{self.first}, {self.second})")
46
47 def make_coordinates(first, second):
48     if is_number(first) and is_number(second) and first > 0 and second > 0:
49         coordinates = Coordinates(first, second)
50         return coordinates
51     else:
52         raise ValueError
53
54
55 if __name__ == '__main__':
56     p = make_coordinates(2, 10)
57     p.display()
58     p.rangecheck(5)
59     p.rangecheck(10)
60
61 if __name__ == '__main__':
```

Рисунок 4.1.3 – Проработка программы

A screenshot of a Python IDE's console window. The window title is 'Run: Ind_1 x'. The console output shows the following text: 'C:\Users\kuvsh\Desktop\СКФУ\2_4_семестр\Основы Программной инженерии', 'First value: 2', 'Second value: 10', '5 принадлежит диапазону [2,10)', '10 не принадлежит диапазону [2,10)', '2 принадлежит диапазону [2,10)', and 'Process finished with exit code 0'. The left sidebar shows standard IDE icons for running, debugging, and file management.

```
Run: Ind_1 x
"C:\Users\kuvsh\Desktop\СКФУ\2_4_семестр\Основы Программной инженерии
First value: 2
Second value: 10
5 принадлежит диапазону [2,10)
10 не принадлежит диапазону [2,10)
2 принадлежит диапазону [2,10)
Process finished with exit code 0
```

Рисунок 4.1.4 – Результат работы программы

Задание 2

Составить программу с использованием классов и объектов для решения задачи. Во всех заданиях, помимо указанных в задании операций, обязательно должны быть реализованы следующие методы:

метод инициализации `__init__` ;

ввод с клавиатуры `read` ;

вывод на экран `display` .

Номер варианта необходимо уточнить у преподавателя. В раздел программы, начинающийся после инструкции `if __name__ == '__main__':` добавить код, демонстрирующий возможности разработанного класса.

8. Создать класс `Time` для работы со временем в формате «час:минута:секунда». Класс должен включать в себя не менее четырех функций инициализации: числами, строкой (например, «23:59:59»), секундами и временем. Обязательными операциями являются: вычисление разницы между двумя моментами времени в секундах, сложение времени и заданного количества секунд, вычитание из времени заданного количества секунд, сравнение моментов времени, перевод в секунды, перевод в минуты (с округлением до целой минуты).

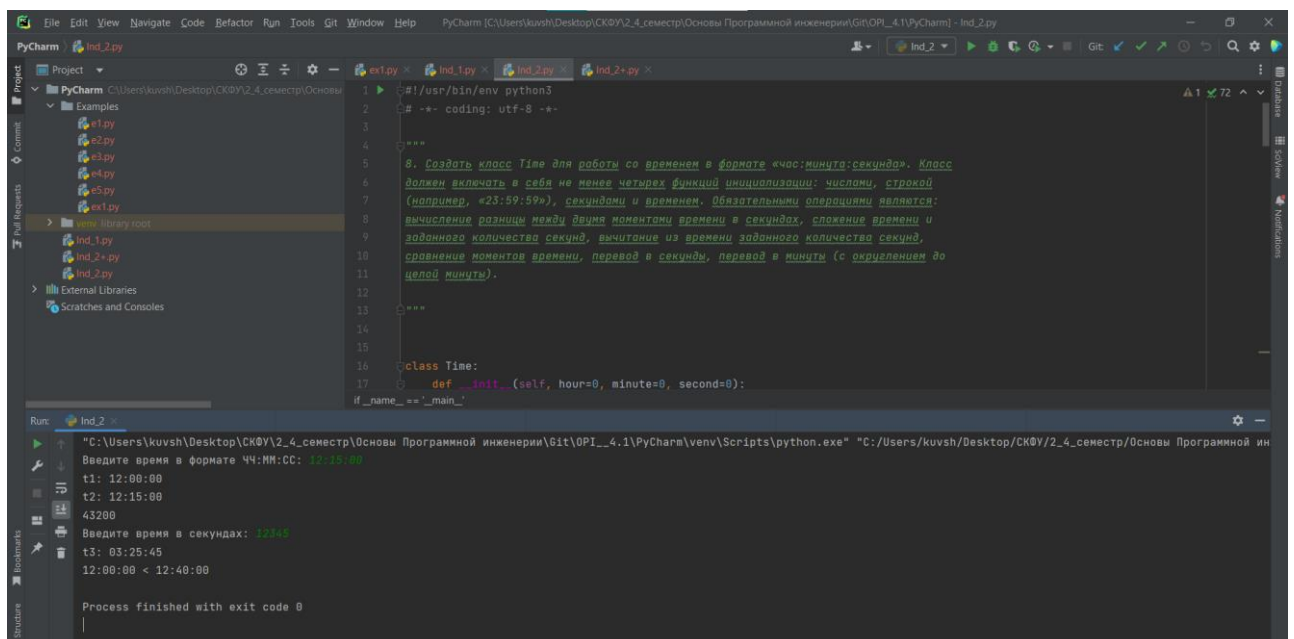
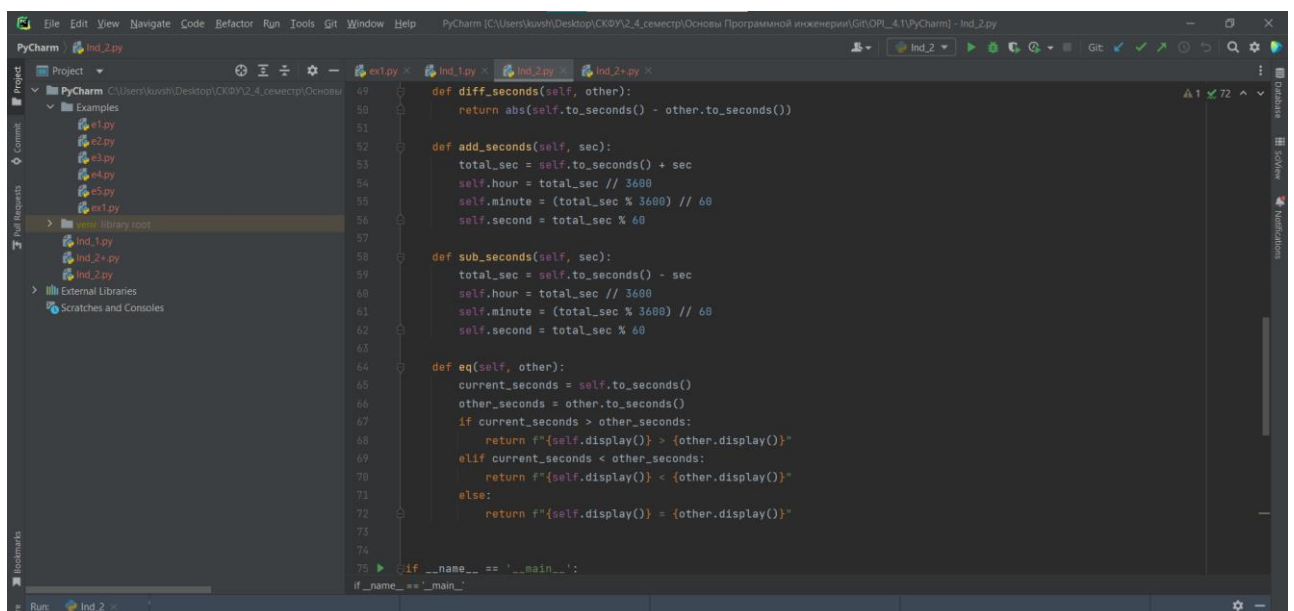
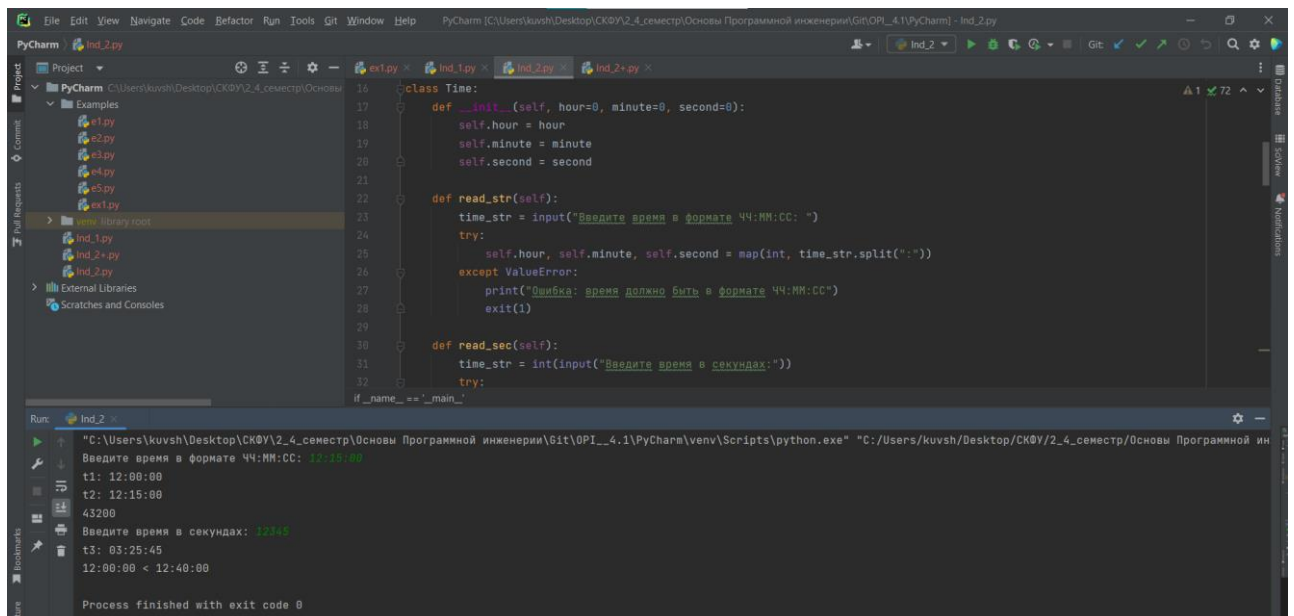
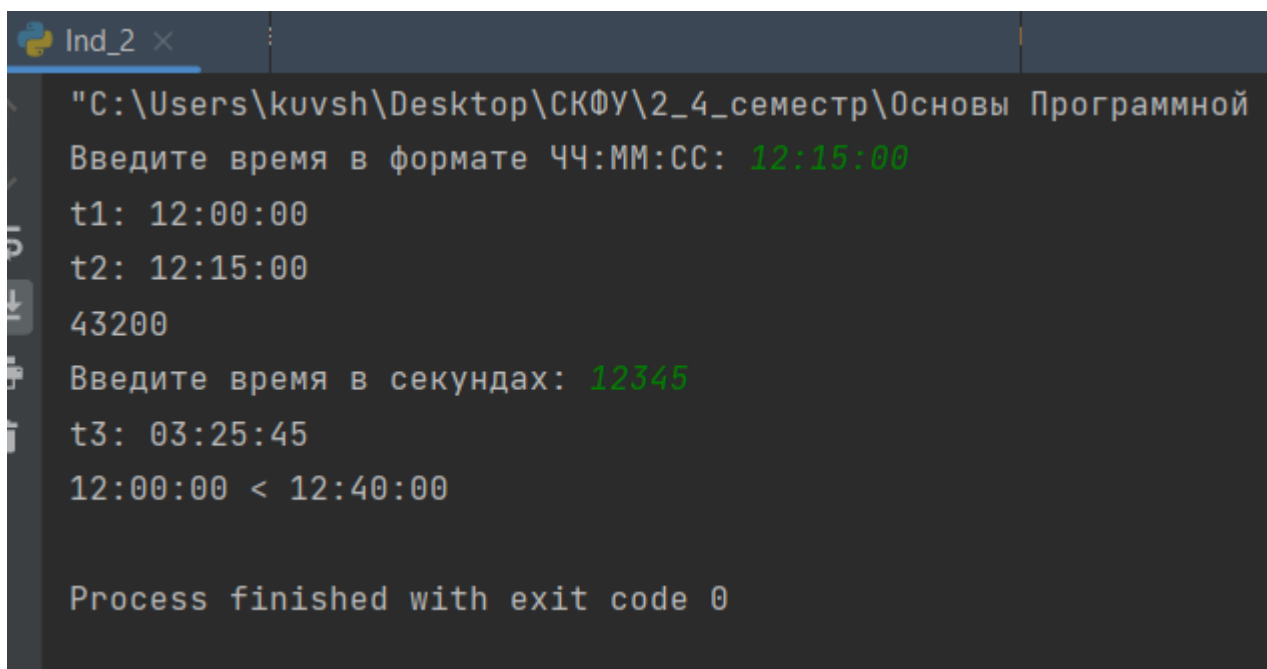


Рисунок 4.1.5 – Проработка программы



```
"C:\Users\kuvsh\Desktop\СКОУ\2_4_семестр\Основы Программной
Введите время в формате ЧЧ:ММ:СС: 12:15:00
t1: 12:00:00
t2: 12:15:00
43200
Введите время в секундах: 12345
t3: 03:25:45
12:00:00 < 12:40:00

Process finished with exit code 0
```

Рисунок 4.1.6 – Результат работы программы

9. Зафиксируйте сделанные изменения в репозитории.
10. Выполните слияние ветки для разработки с веткой main (master).
11. Отправьте сделанные изменения на сервер GitHub.

Контрольные вопросы

1. Как осуществляется объявление класса в языке Python?

Объявление класса в Python осуществляется с помощью ключевого слова `class`, за которым следует имя класса и двоеточие. Например, `class MyClass:`

2. Чем атрибуты класса отличаются от атрибутов экземпляра?

Атрибуты класса принадлежат классу в целом и доступны всем его экземплярам. Атрибуты экземпляра, наоборот, принадлежат конкретному экземпляру класса и могут быть разными у разных объектов.

3. Каково назначение методов класса?

Методы класса являются функциями, которые связаны с классом и могут быть вызваны у его экземпляров. Они часто используются для работы с атрибутами экземпляра, их изменения или чтения.

4. Для чего предназначен метод `__init__()` класса?

`__init__()` - это метод класса, который вызывается при создании экземпляра объекта. Он используется для установки начальных значений атрибутов объекта.

5. Каково назначение `self` ?

`self` - это первый параметр всех методов класса в Python. Данный параметр используется для доступа и изменения атрибутов объекта, от которого вызывается метод.

6. Как добавить атрибуты в класс?

Для добавления атрибутов в класс нужно просто создать новые атрибуты в методе класса или за его пределами, используя имя класса, например: `MyClass.new_attribute = "value"`.

7. Как осуществляется управление доступом к методам и атрибутам в языке Python?

Управление доступом к методам и атрибутам в Python осуществляется с помощью модификаторов доступа: `public`, `private` и `protected`. В Python все атрибуты и методы класса по умолчанию являются публичными, т.е. доступ к ним возможен из любого места программы. Для создания приватных атрибутов и методов используется синтаксис с двойным подчеркиванием (`__`), а для создания защищенных атрибутов и методов одинарным подчеркиванием (`_`).

8. Каково назначение функции `isinstance` ?

Функция `isinstance()` возвращает `True`, если переданный объект является экземпляром указанного класса или его потомком. Она используется для проверки типа объекта в Python.

