**АНАЛИЗ ДАННЫХ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ [in GameDev]**

Отчет по лабораторной работе #1 выполнил(а):

* Засыпкина Елена Юрьевна
* РИ210940 Отметка о выполнении заданий (заполняется студентом):

| **Задание** | **Выполнение** | **Баллы** |
| --- | --- | --- |
| Задание 1 | \* | 60 |
| Задание 2 | \* | 20 |
| Задание 3 | \* | 20 |

знак "\*" - задание выполнено; знак "#" - задание не выполнено;

Работу проверили:

* к.т.н., доцент Денисов Д.В.
* к.э.н., доцент Панов М.А.
* ст. преп., Фадеев В.О.

[N|Solid](https://nodesource.com/products/nsolid)

Структура отчета

* Данные о работе: название работы, фио, группа, выполненные задания.
* Цель работы.
* Задание 1.
* Код реализации выполнения задания. Визуализация результатов выполнения (если применимо).
* Задание 2.
* Код реализации выполнения задания. Визуализация результатов выполнения (если применимо).
* Задание 3.
* Код реализации выполнения задания. Визуализация результатов выполнения (если применимо).
* Выводы.
* ✨Magic ✨

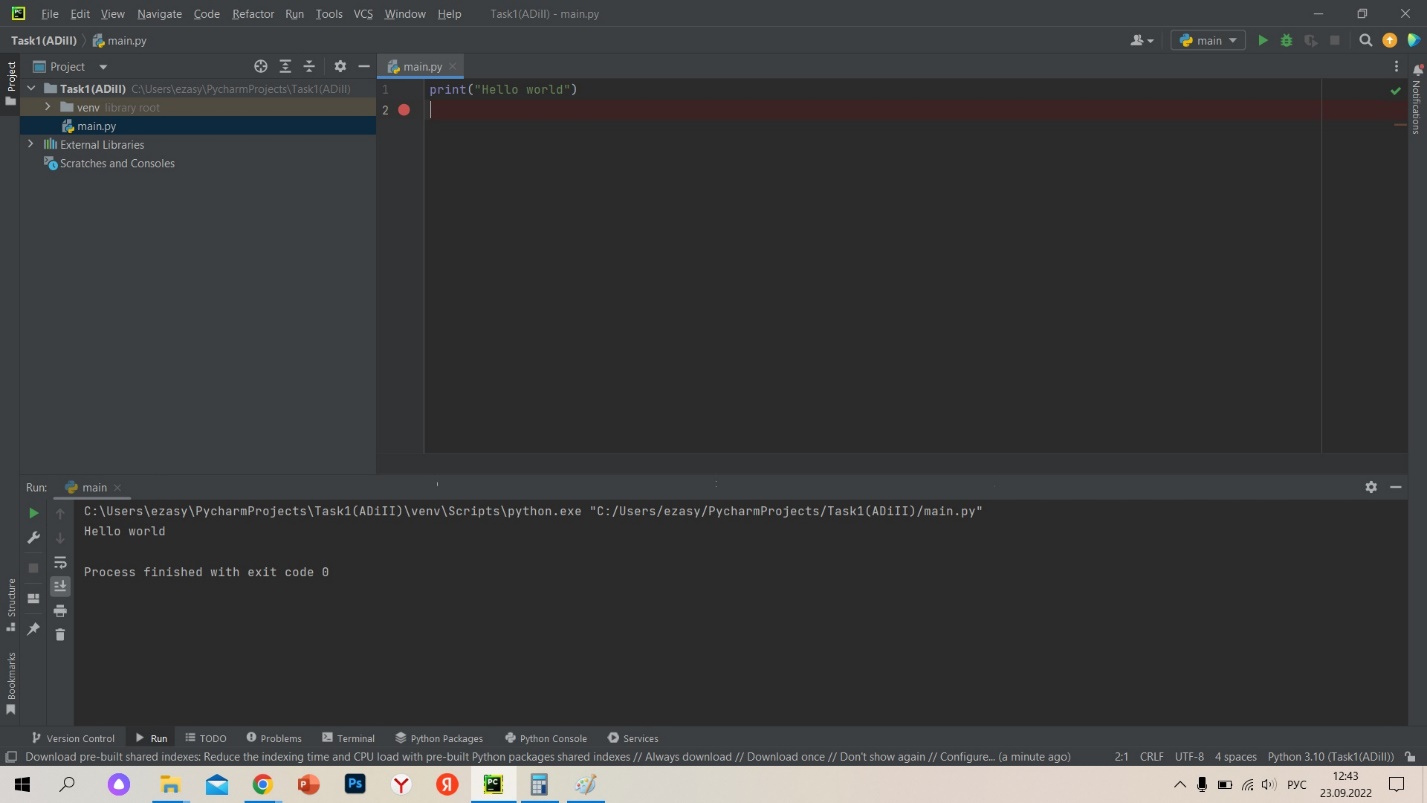
**Цель работы**

Ознакомиться с основными операторами зыка Python на примере реализации линейной регрессии.

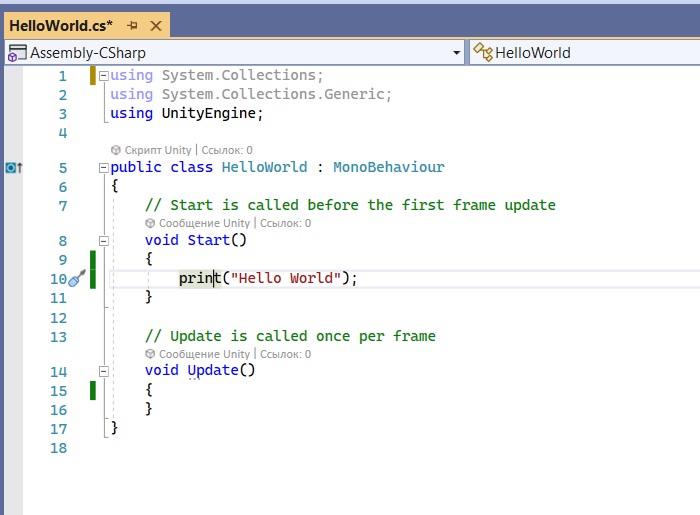
**Задание 1**

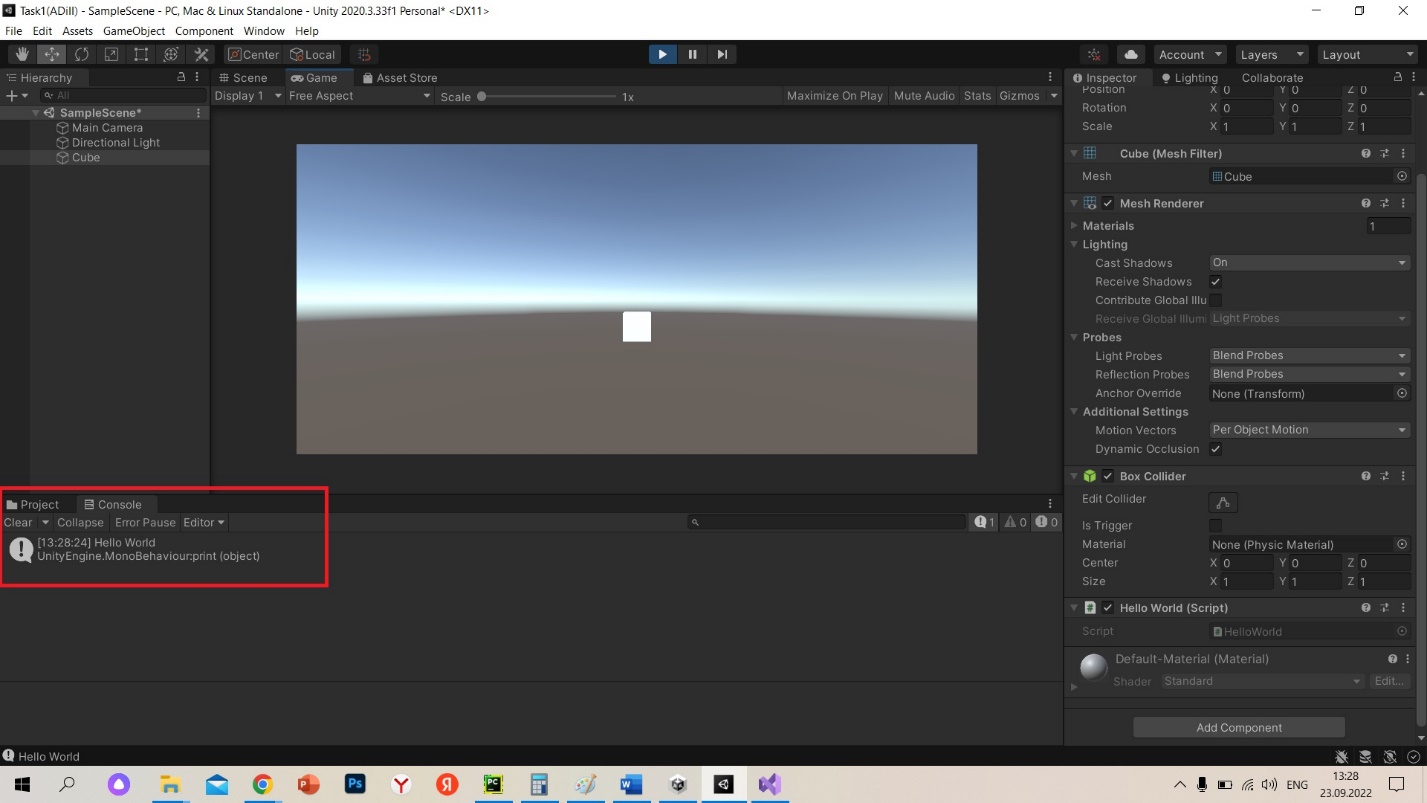
**Написать программы Hello world на Python и Unity**

Для Python я решила использовать PyCharm. Создала новый проект и написала нужную программу:

[](https://user-images.githubusercontent.com/102030455/191915927-ef3f27a3-0bfb-4d6f-b19f-6a0ca4f79613.jpg)

В Unity примерно то же самое. Создала проект. В проекте создала куб и привязала к нему скрипт, который вывод сообщение в консоль:

[](https://user-images.githubusercontent.com/102030455/191921564-42d7977f-dcea-4d33-89af-73307954b88d.jpg)

[](https://user-images.githubusercontent.com/102030455/191921586-11f19607-2e9d-41a8-b14f-989d2d7b0684.jpg)

**Задание 2**

**Пошагово выполнить каждый пункт раздела "ход работы" с описанием и примерами реализации задач**

Ход работы:

* Произвести подготовку данных для работы с алгоритмом линейной регрессии. 10 видов данных были установлены случайным образом, и данные находились в линейной зависимости. Данные преобразуются в формат массива, чтобы их можно было вычислить напрямую при использовании умножения и сложения.

import numpy as np

import matplotlib.pyplot as plt

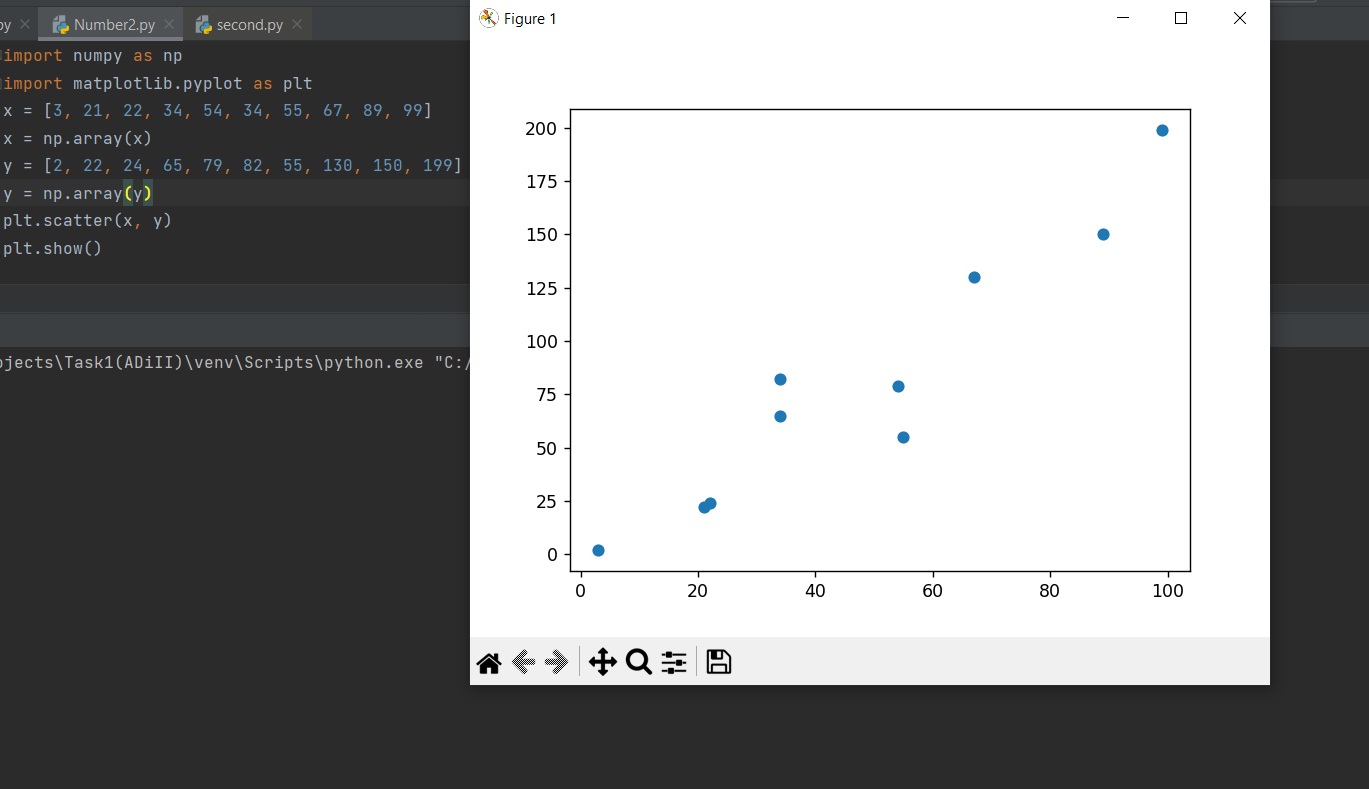
x = [3, 21, 22, 34, 54, 34, 55, 67, 89, 99]

x = np.array(x)

y = [2, 22, 24, 65, 79, 82, 55, 130, 150, 199]

y = np.array(y)

plt.scatter(x, y)

[](https://user-images.githubusercontent.com/102030455/192087815-1950eac2-d3f0-4d2b-9a5b-16cc50a6a015.jpg)

* Определите связанные функции. Функция модели: определяет модель линейной регрессии wx+b. Функция потерь: функция потерь среднеквадратичной ошибки. Функция оптимизации: метод градиентного спуска для нахождения частных производных w и b.

def model(a, b, x):

return a \* x + b

def lossFunction(a, b, x, y):

num = len(x)

prediction = model(a, b, x)

return (0.5 / num) \* (np.square(prediction - y)).sum()

def optimize(a, b, x, y, Lr):

num = len(x)

prediction = model(a, b, x)

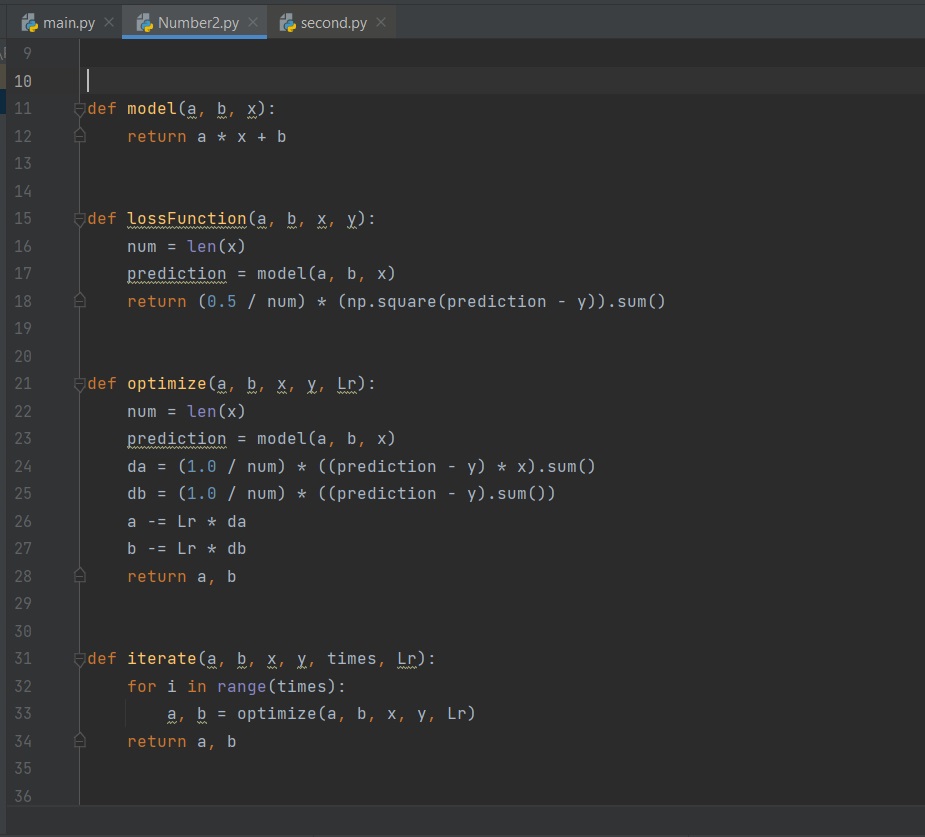
da = (1.0 / num) \* ((prediction - y) \* x).sum()

db = (1.0 / num) \* ((prediction - y).sum())

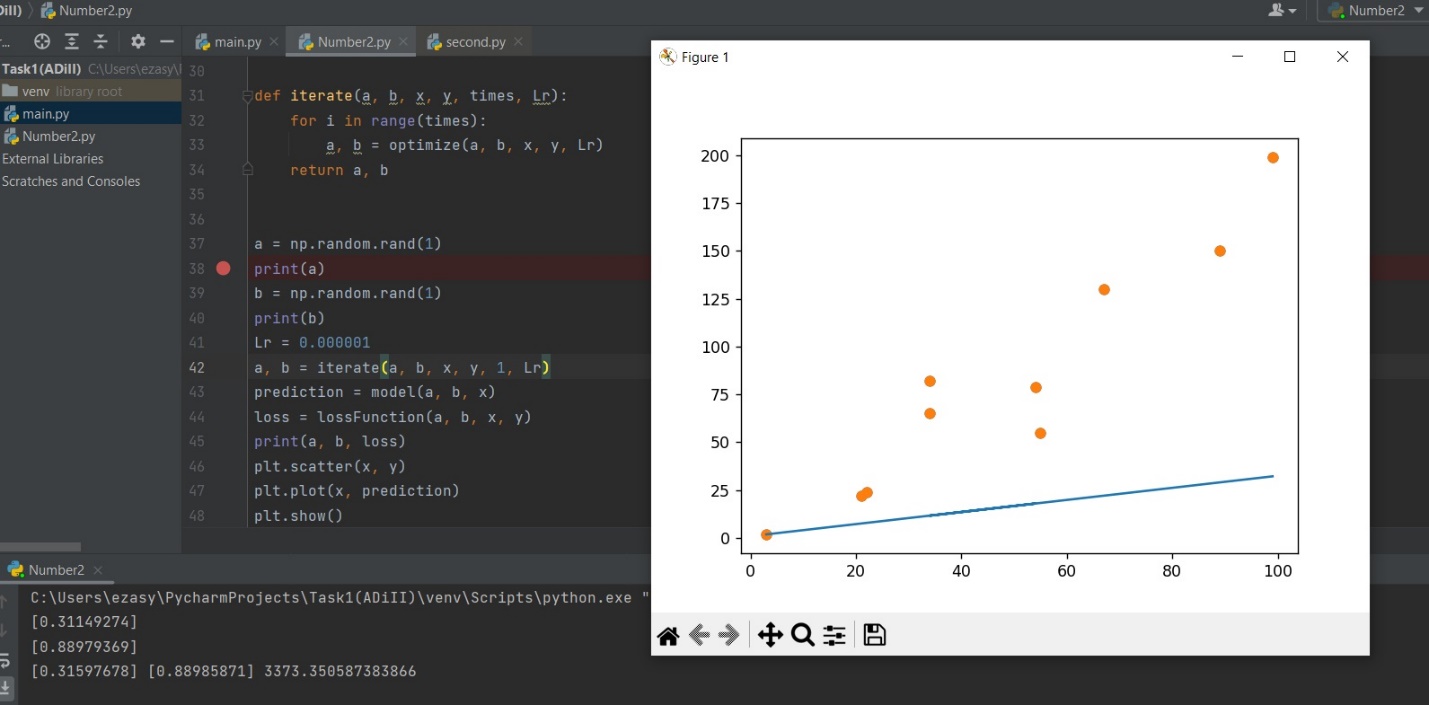
a -= Lr \* da

b -= Lr \* db

return a, b

[](https://user-images.githubusercontent.com/102030455/192087856-c5200bc3-5d63-41f6-9a8c-8d7d5c7f8667.jpg)

-Начать итерацию

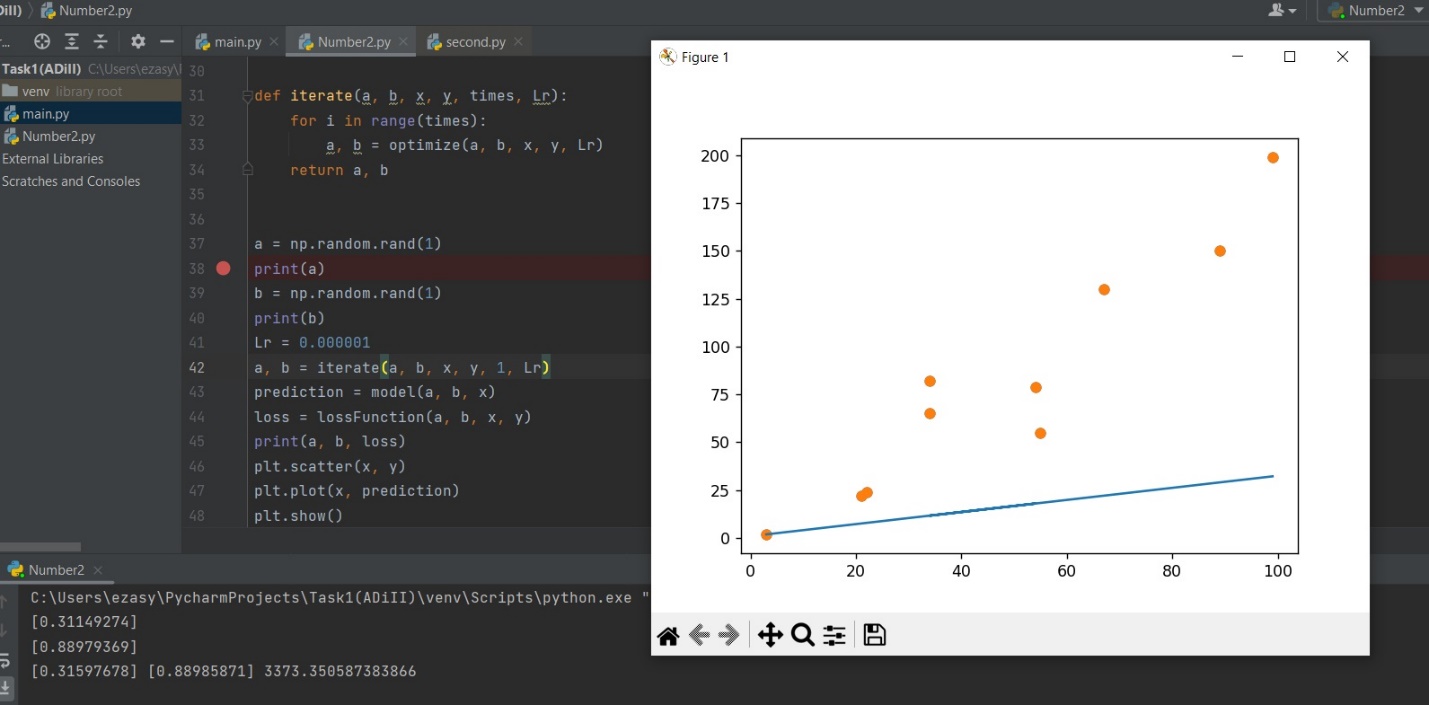
[](https://user-images.githubusercontent.com/102030455/192089334-ae88fb99-e85e-4913-808d-2bf459ea597c.jpg)

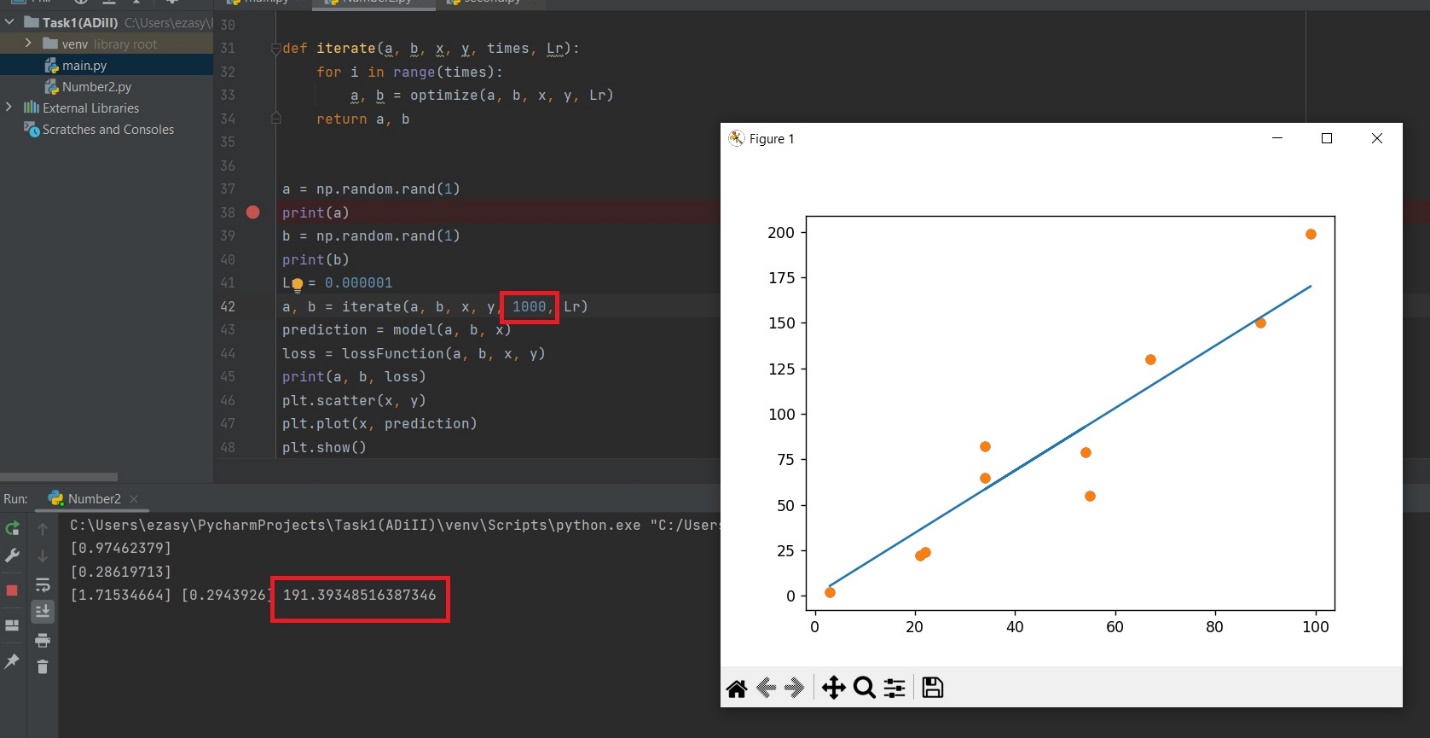
**Задание 3**

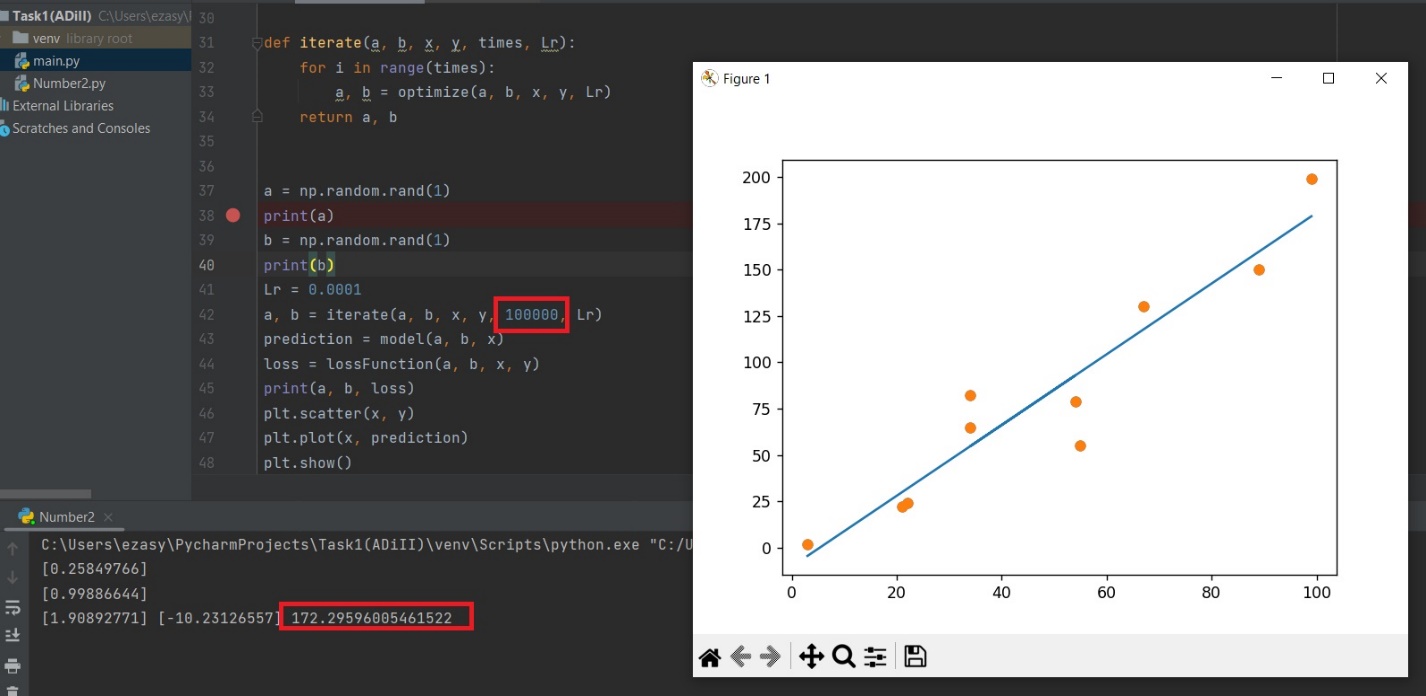
**Должна ли величина loss стремиться к нулю при изменении исходных данных? Ответьте на вопрос, приведите пример выполнения кода, который подтверждает ваш ответ.**

**Какова роль параметра Lr? Ответьте на вопрос, приведите пример выполнения кода, который подтверждает ваш ответ. В качестве эксперимента можете изменить значение параметра.**

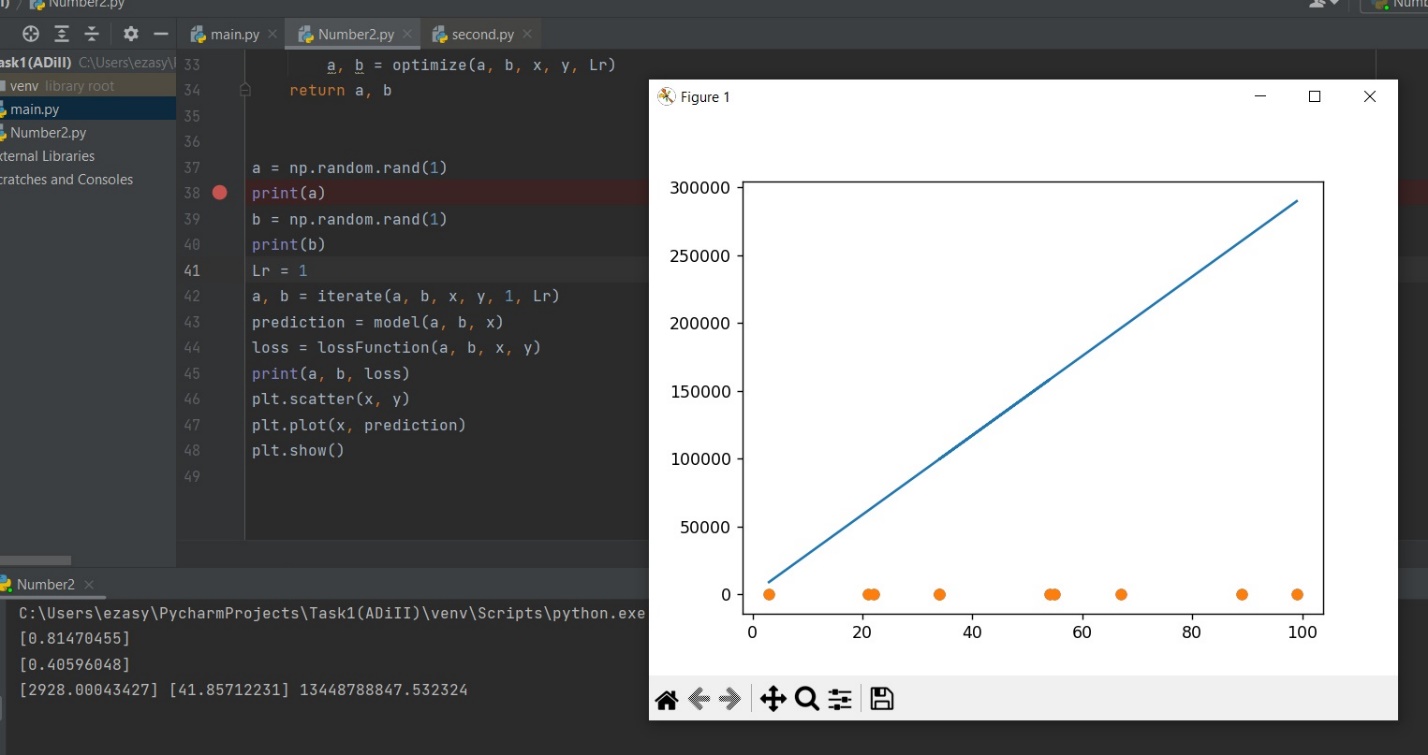
1. Да, должна. Например, если увеличить количество итерций, то loss станет меньше:

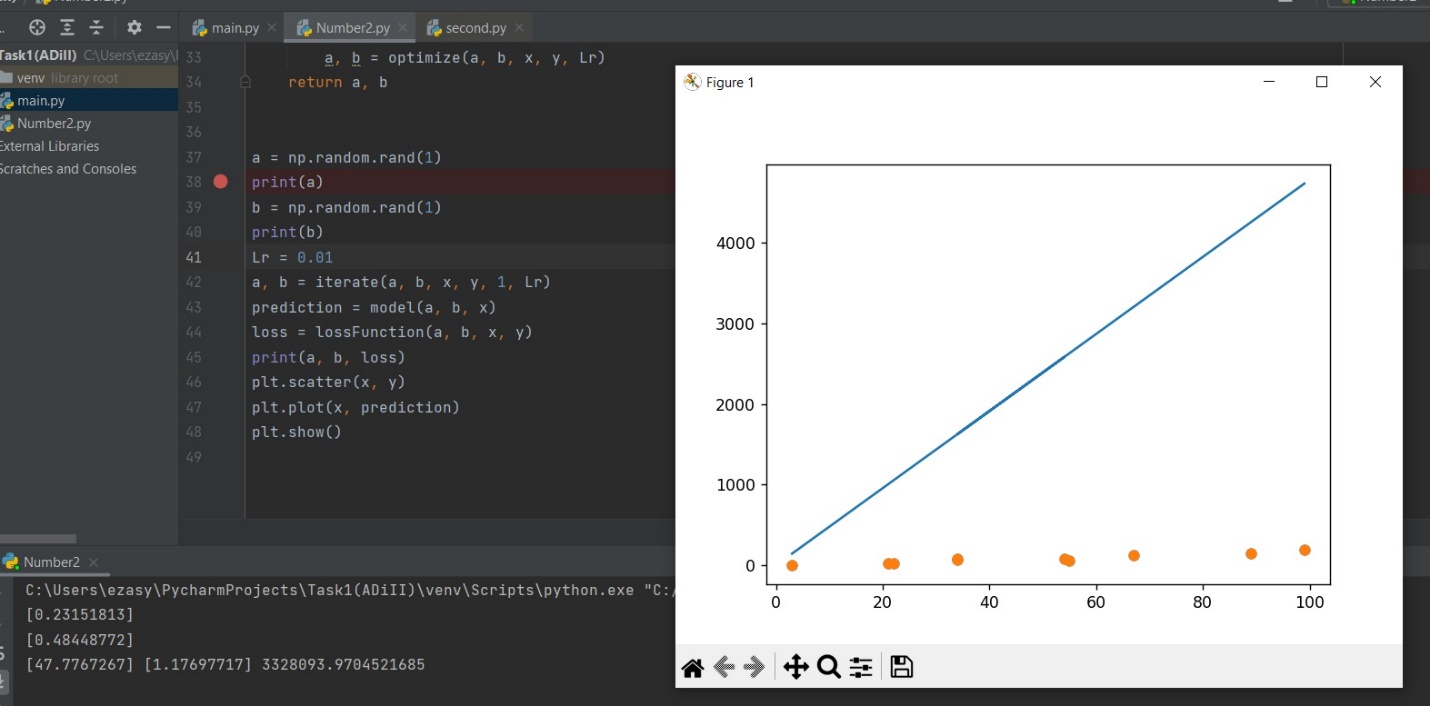
[](https://user-images.githubusercontent.com/102030455/192090382-608f0be0-3799-4331-a90d-2afc6a92509f.jpg)

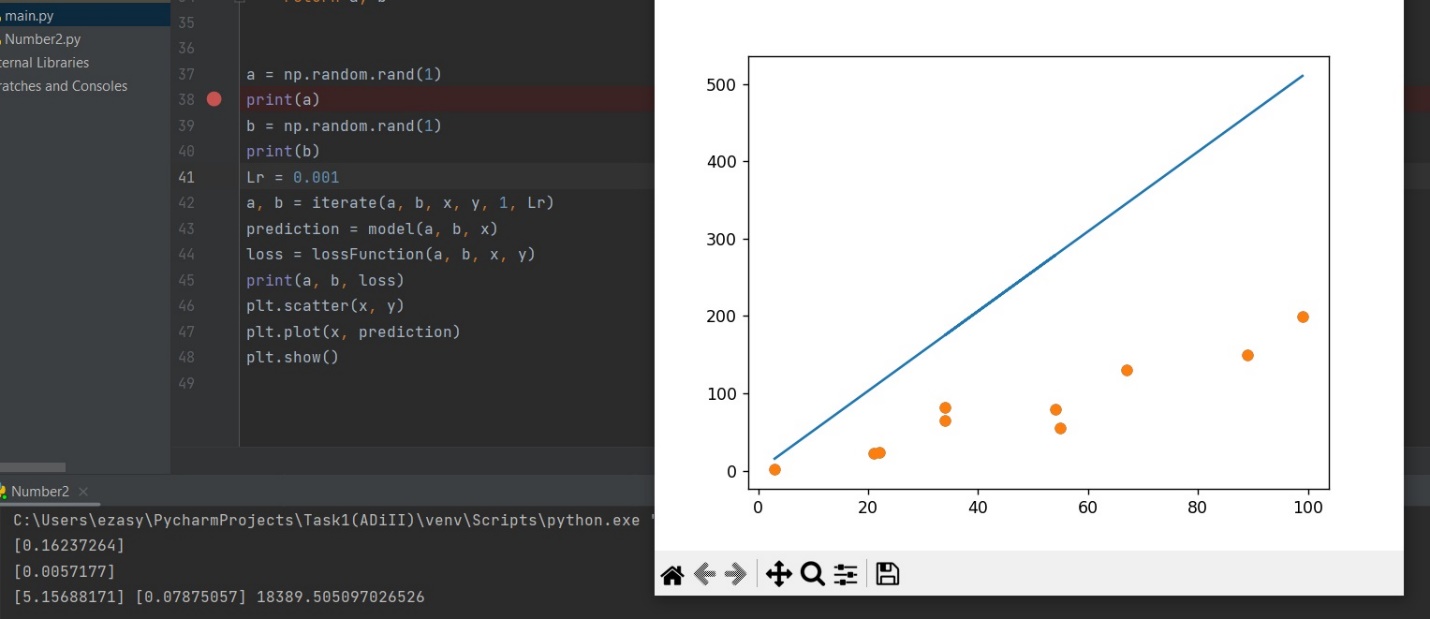
[](https://user-images.githubusercontent.com/102030455/192090388-69fc8094-d071-46b3-b49d-38a061078730.jpg)

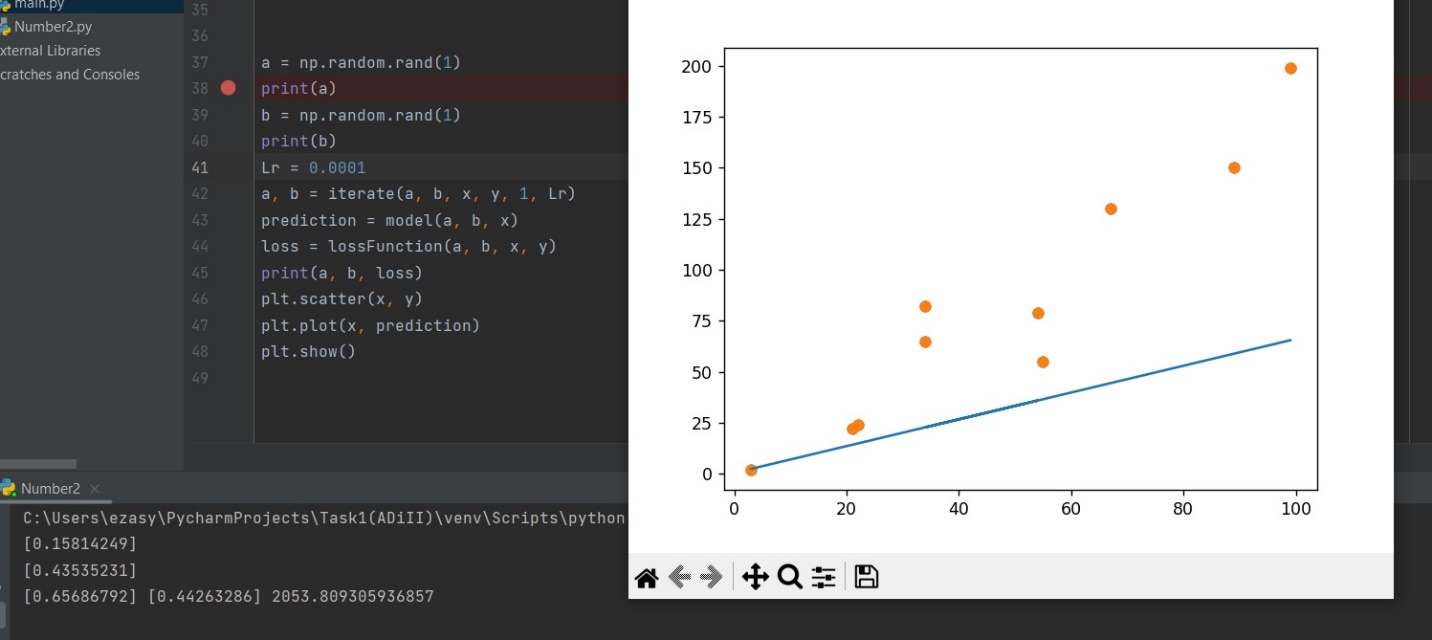
[](https://user-images.githubusercontent.com/102030455/192090419-3adbcc6d-a93c-4fc8-ac75-9eca66fc5b98.jpg)

1. Lr влияет на наклон графика (прямой), при использовании функции optimize. Чем меньше Lr, тем больше наклонен график:

Lr=1: [](https://user-images.githubusercontent.com/102030455/192091651-ee625ae7-d521-4b87-8ba5-f7897e56e782.jpg)

Lr=0.01: [](https://user-images.githubusercontent.com/102030455/192091656-a8b0ee71-0d9b-404a-960b-e05c0b337675.jpg)

Lr=0.001: [](https://user-images.githubusercontent.com/102030455/192091658-79e191e6-28d5-4694-90f1-d0df5d23ee7d.jpg)

L=0.0001: [](https://user-images.githubusercontent.com/102030455/192091660-bbb961b6-52be-4664-98f3-9c3f79d0def1.jpg)

**Выводы**

Научилась делать вывод в консоль, используя Python и Unity (C#). Также был проведен небольшой анализ кода для ответа на вопросы.

| **Plugin** | **README** |
| --- | --- |
| Dropbox | [plugins/dropbox/README.md][PlDb] |
| GitHub | [plugins/github/README.md][PlGh] |
| Google Drive | [plugins/googledrive/README.md][PlGd] |
| OneDrive | [plugins/onedrive/README.md][PlOd] |
| Medium | [plugins/medium/README.md][PlMe] |
| Google Analytics | [plugins/googleanalytics/README.md][PlGa] |

**Powered by**

**BigDigital Team: Denisov | Fadeev | Panov**