

Санкт-Петербургский государственный университет

Прикладная математика и информатика

Отчет по учебной практике (научно-исследовательской работе) (семестр 1)
Решение финансовых задач с помощью Python

Выполнил:

Подлеснов Яков Сергеевич, группа 20.Б04-мм



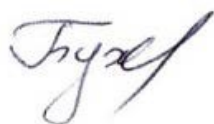
Научный руководитель:

Канд. ф-м. н., ст. преподаватель

Бухвалова Вера Вацлавовна.

Кафедра исследования операций

*Работа выполнена на качественном уровне и может быть зачтена с оценкой
отлично (А).*



Санкт-Петербург

2020

Введение

Цель моей учебной практики состояла в том, чтобы изучить основы финансовой математики, а конкретнее, разобраться в финансовых вычислениях, изучить виды процентов: простые и сложные. Данные знания мне помогут в будущем выбрать, где взять кредит, а также открыть выгодный для меня вклад в банке.

Изученный материал + решенные задачи

Глава 1. Процентные вычисления

В первой главе я познакомился с такими понятиями, как процент, портфель ценных бумаг, начисление налогов, социально-экономические показатели, методы ценообразования. Ключевыми формулами данной главы, по моему мнению, являлись:

1) Число, которое составляет n % от числа A . Формула (1.1).

$$x = \frac{A \times n}{100} \quad (1.1)$$

2) Определить число, n % которого равны A . Формула (1.2).

$$x = \frac{A \times 100}{n} \quad (1.2)$$

3) Определить, сколько процентов от числа B составляет число A . Формула (1.3).

$$x = \frac{A \times 100\%}{B} \quad (1.3)$$

Задачи, которые были решены для закрепление теории из 1 главы.

Задача 1.

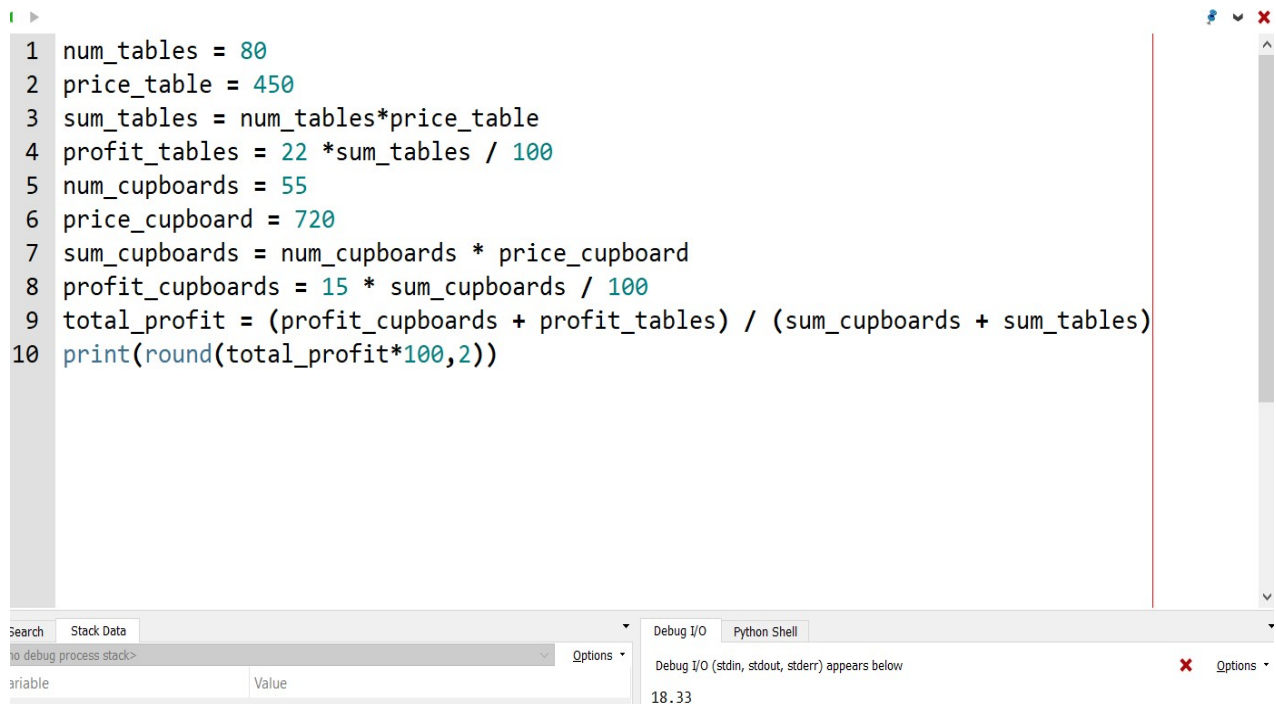
Торговая компания купила 80 столов по цене 450 руб. за стол и 55 шкафов по цене 720 руб. за шкаф. Компания продала всю эту мебель, получив от продажи столов 22% прибыли, а от продажи шкафов 15% прибыли. Сколько процентов прибыли получила компания от продажи всей мебели?

Для решение этой задачи достаточно использовать формулу (1.1) для нахождения прибыли от продажи шкафов и столов, и формулу (1.3) для нахождения прибыли от всей всей продажи.

Программный код, решающий задачу:

```
num_tables = 80
price_table = 450
sum_tables = num_tables*price_table
profit_tables = 22 *sum_tables / 100
num_cupboards = 55
price_cupboard = 720
sum_cupboards = num_cupboards * price_cupboard
profit_cupboards = 15 * sum_cupboards / 100
total_profit = (profit_cupboards + profit_tables) / (sum_cupboards +
sum_tables)
print(round(total_profit*100,2))
```

Скриншот ответом:



The screenshot shows a Python IDE with the following code in the editor:

```
1 num_tables = 80
2 price_table = 450
3 sum_tables = num_tables*price_table
4 profit_tables = 22 *sum_tables / 100
5 num_cupboards = 55
6 price_cupboard = 720
7 sum_cupboards = num_cupboards * price_cupboard
8 profit_cupboards = 15 * sum_cupboards / 100
9 total_profit = (profit_cupboards + profit_tables) / (sum_cupboards + sum_tables)
10 print(round(total_profit*100,2))
```

The output console at the bottom right shows the result of the execution:

```
Debug I/O Python Shell
Debug I/O (stdin, stdout, stderr) appears below
18.33
```

Задача 2.

Портфель ценных бумаг включает 20 облигаций, номинальная цена которых 50 руб., и 70 облигаций, номинальная цена которых 100 руб. Облигации первого вида приносят владельцу 12% дохода в год, второго — 9% дохода в год. Вычислите, сколько процентов годового дохода получит владелец данного портфеля от всех входящих в него облигаций.

Для данной задачи аналогично 1 воспользуемся формулой (1.1) для подсчёта годового дохода от 1 и 2 видов облигаций. И формулой (1.3) для нахождения годового дохода от всего портфеля ценных бумаг.

Программный код, решающий задачу:

```
num1_obligations = 20
```

```
price1_obligations = 50
```

```
num2_obligations = 70
```

```
price2_obligations = 100
```

```
sum1= num1_obligations*price1_obligations
```

```
sum2=num2_obligations*price2_obligations
```

```
r1 = 12
```

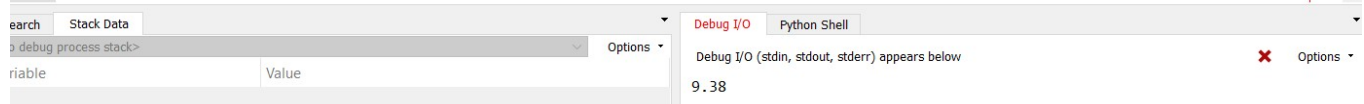
```
r2 = 9
```

```
profit = (sum1*r1/100+sum2*r2/100)/(sum1+sum2)
```

```
print(round(profit*100,2))
```

Скриншот с ответом:

```
1 num1_obligations = 20
2 price1_obligations = 50
3 num2_obligations = 70
4 price2_obligations = 100
5 sum1= num1_obligations*price1_obligations
6 sum2=num2_obligations*price2_obligations
7 r1 = 12
8 r2 = 9
9 profit = (sum1*r1/100+sum2*r2/100)/(sum1+sum2)
10 print(round(profit*100,2))
```



Задача 3.

В России установлена следующая цена на садовую мебель ТЭРНО: складной стол — 999 руб., складной стул — 799 руб. Цена этих же товаров в США — \$29.99 и \$24.99 соответственно. Курс доллара на момент сравнения составляет 27.80 руб. В какой стране эти товары дороже и на сколько процентов?

Для данной задачи, во-первых, нужно перевести доллары в рубли или наоборот. Во-вторых, при написании программного кода нужно вспомнить про 2 случая: 1 – дороже в РФ, 2 – дороже в США. В-третьих, после того как узнали, где цена выше, применим формулу (1.3) для подсчёта, на сколько процентов товар дороже.

Программный код, решающий задачу:

```
tables_ru = 999
```

```
chairs_ru = 799
```

```
course = 27.8
```

```
tables_usa = 29.99
```

```
chairs_usa = 24.99
```

```
tables_usa_inrubles = tables_usa * course
```

```
chairs_usa_inrubles = chairs_usa * course
```

```
if tables_usa_inrubles > tables_ru:
```

```

q = round(((tables_usa_inrubles/tables_ru -1)*100,2)

print('Tables are',q,'percent more expensive in the USA than in Russia')

else:

q = round(((tables_ru / tables_usa_inrubles-1)*100,2)

print('Tables are',q,'percent more expensive in Russia than in the USA')

if chairs_usa_inrubles > chairs_ru:

q = round(((chairs_usa_inrubles/chairs_ru -1)*100,2)

print('Chairs are',q,'percent more expensive in the USA than in Russia.')

else:

q = round(((chairs_ru / chairs_usa_inrubles-1)*100,2)

print('Chairs are',q,'percent more expensive in Russia than in the USA.')

```

Скриншот с ответом:

```

1 tables_ru = 999
2 chairs_ru = 799
3 course = 27.8
4 tables_usa = 29.99
5 chairs_usa = 24.99
6 tables_usa_inrubles = tables_usa * course
7 chairs_usa_inrubles = chairs_usa * course
8 if tables_usa_inrubles > tables_ru:
9     q = round(((tables_usa_inrubles/tables_ru -1)*100,2)
10     print('Tables are',q,'percent more expensive in the USA than in Russia')
11 else:
12     q = round(((tables_ru / tables_usa_inrubles-1)*100,2)
13     print('Tables are',q,'percent more expensive in Russia than in the USA')
14 if chairs_usa_inrubles > chairs_ru:
15     q = round(((chairs_usa_inrubles/chairs_ru -1)*100,2)
16     print('Chairs are',q,'percent more expensive in the USA than in Russia.')
17 else:
18     q = round(((chairs_ru / chairs_usa_inrubles-1)*100,2)
19     print('Chairs are',q,'percent more expensive in Russia than in the USA.')
20

```

Search	Stack Data	Options	Debug I/O	Python Shell
no debug process stack>			Debug I/O (stdin, stdout, stderr) appears below	
variable	Value		<p>Tables are 19.82 percent more expensive in Russia than in the USA</p> <p>Chairs are 15.01 percent more expensive in Russia than in the USA.</p>	

Глава 2. Простые проценты.

Во второй главе мне встретились такие термины, как простые проценты, перевод раз в проценты и обратно, банковский депозит под простые проценты, процентный пункт, векселя, учет векселей, простой дисконт, приведение ценности денег к одному моменту времени, эквивалентность учетной и

процентой ставки, влияние инфляции на ставку процента. В данной части книги представлено большое кол-во формул и тяжело выделить какие-то основные, как в 1 главе, однако исключением будет первая формула, представленная в самом начале и выражающая наращенную сумму при начислении простых процентов, при условии, что $t > 0$ (число периодов). Формула (2.1). Другие формулы, представленные в данной главе буду иллюстрировать при объяснении решенных мною задач.

$$S = P(1 + rt) . \quad (2.1)$$

Задача 1.

В банк, выплачивающий 6% простых годовых, положили 6 000 руб. Через сколько лет на счете будет 6 540 руб.?

Для решения данной задачи достаточно воспользоваться формулой (2.1) и выразить t - окончательная формула представлена в программном коде.

(n – тоже самое, что и t в формуле (2.1)).

Программный код, решающий задачу:

```
r = 6
```

```
P = 6000
```

```
S = 6540
```

```
n = round((S - P) / (P*(r/100)),2)
```

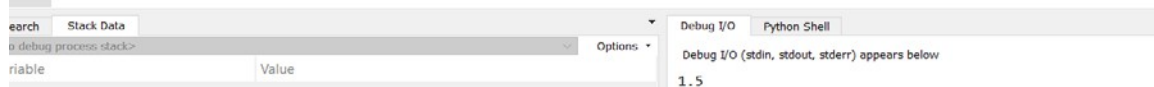
```
print(n)
```

Скриншот с ответом:


```

1 r = 6
2 P = 6000
3 S = 6540
4 n = round((S - P) / (P*(r/100)),2)
5 print(n)
6

```



Задача 2.

Фирма планирует приобрести новые помещения, за которые она должна заплатить \$120 000. Фирма имеет два предложения. По первому предложению фирма должна выплатить эту сумму за 3 года, выплачивая в конце каждого года по \$40 000. По второму предложению фирма должна заплатить сразу \$30 000, а остальные \$90 000 погашать равными суммами каждые полгода, выплатив весь долг к концу третьего года. Какое предложение выгоднее для фирмы? На деньги начисляются 6% годовых (простых).

В данной задаче условие нам подсказывает, что нужно воспользоваться формулой (2.2) приведение ценности к одному моменту времени.

$$P = \frac{S}{1 + rt} . \quad (2.2)$$

Замечу, что для каждого периода времени мы должны написать отдельную формулу, для удобства я создал цикл `while`, который упрощает работу в подсчёте, а также сокращает время написания программного кода.

Программный код, решающий задачу:

n = 3

i = -n + 1 #сделал отдельные переменные i и j для циклов, чтобы не писать длинные формулы для P1 и P2


```

j = -n + 0.5
r = 0.06
P1 = 0
S1 = 40000
S2_1 = 30000
P2 = S2_1
S2_2 = 90000 // 6
while i != 1:
    P1 += S1/(1+r*(n+i))
    i += 1
while j != 0.5:
    P2 += S2_2/(1+r*(n+j))
    j += 0.5
if (P1 < P2):
    print("The first offer is more profitable.")
else:
    print("The second offer is more profitable.")

```

Скриншот с ответом:

```

1  n = 3
2  i = -n + 1#сделал отдельные переменные i и j для циклов,чтобы не писать длинные
   формулы для P1 и P2
3  j = -n + 0.5
4  r = 0.06
5  P1 = 0
6  S1 = 40000
7  S2_1 = 30000
8  P2 = S2_1
9  S2_2 = 90000 // 6
10 while i != 1:
11     P1 += S1/(1+r*(n+i))
12     i += 1
13 while j != 0.5:
14     P2 += S2_2/(1+r*(n+j))
15     j += 0.5
16 if (P1 < P2):
17     print("The first offer is more profitable.")
18 else:
19     print("The second offer is more profitable.")

```

Search Stack Data

Debug I/O Python Shell

Debug I/O (stdin, stdout, stderr) appears below

The first offer is more profitable.

Задача 3.

Г-н Гаврилов должен выплатить г-ну Серову 20 000 руб. в следующие сроки: 5 000 руб. через 2 года, 5 000 руб. через 3 года и еще 10 000 руб. через 5 лет, считая от настоящего момента. Г-н Гаврилов предложил изменить контракт, обязавшись уплатить 10 000 руб. через 3 года и еще 10 000 руб. через 4 года от настоящего момента. Эквивалентны ли эти контракты, если на деньги начисляются 5% годовых (простых)? Если контракты не эквивалентны, то какой из них выгоднее для г-на Серова?

Для данной задачи достаточно продисконтировать суммы (P_1 , P_2 , P_3 – первый контракт, P_{1_1} , P_{2_2} – второй контракт) по формуле из (рис.5) и применить условие $P_1 + P_2 + P_3 = P_{1_1} + P_{2_2}$, если данное условие выполнилось – эквивалентны, в противном случае – выбираем какое меньше (выгоднее).

Программный код, решающий задачу:

n1 = 2

s1 = 5000

n2 = 3

s2 = 5000

n3 = 5

s3 = 10000

n1_2 = 3

s1_2 = 10000

```

n2_2 = 4
s2_2 = 10000
r = 0.05
p1 = p2 = p3 = p1_2 = p2_2 = 0
p1 = round(s1/(1+n1*r))
p2 = round(s2/(1+n2*r))
p3 = round(s3/(1+n3*r))
p1_2 = round(s1_2/(1+n1_2*r))
p2_2 = round(s2_2/(1+n2_2*r))
if p1 + p2 + p3 == p1_2 + p2_2:
    print("The offers are equivalent.")
elif p1 + p2 + p3 < p1_2 + p2_2:
    print("The first offer is more profitable.")
else:
    print("The second offer is more profitable.")

```

Скриншот с ответом:

```

1  n1 = 2
2  s1 = 5000
3  n2 = 3
4  s2 = 5000
5  n3 = 5
6  s3 = 10000
7  n1_2 = 3
8  s1_2 = 10000
9  n2_2 = 4
10 s2_2 = 10000
11 r = 0.05
12 p1 = p2 = p3 = p1_2 = p2_2 = 0
13 p1 = round(s1/(1+n1*r))
14 p2 = round(s2/(1+n2*r))
15 p3 = round(s3/(1+n3*r))
16 p1_2 = round(s1_2/(1+n1_2*r))
17 p2_2 = round(s2_2/(1+n2_2*r))
18 if p1 + p2 + p3 == p1_2 + p2_2:
19     print("The offers are equivalent.")
20 elif p1 + p2 + p3 < p1_2 + p2_2:
21     print("The first offer is more profitable.")
22 else:
23     print("The second offer is more profitable.")

```

Search Stack Data

no debug process stack> Options

Variable	Value

Debug I/O Python Shell

Debug I/O (stdin, stdout, stderr) appears below

The first offer is more profitable.

Глава 3. Сложные проценты.

Это глава была насыщена такими понятиями, как сложные проценты, непрерывное начисление сложных процентов, учет векселей по сложной учетной ставке, эквивалентность процентных ставок, эффективная процентная ставка, плавающий сложный процент, правило 69 и правило 72.

Аналогично, как и в главе 2 выделяю одну основную формулу из огромного количества формула (3.2), но отмечу, что другие формулы тоже важны. Некоторые из них проиллюстрирую по мере объяснения задач, решенных мною.

Формула, выражающая наращенную сумму при начислении сложных процентов.

$$S = P(1 + r)^t \quad (3.2)$$

Задача 1.

Фермер хочет вложить 30 000 руб., чтобы через 5 лет получить 40 000 руб. Под какую процентную ставку j_{12} он должен вложить свои деньги?

В данной задаче я воспользовался формулой для подсчета суммы, когда начисление сложных процентов производится m раз в году по ставке r/m .

Формула (3.5). Для получения ответа достаточно выразить j_m - окончательная формула представлена в программном коде.

$$S = P \left(1 + \frac{j_m}{m} \right)^{tm} .$$

(3.5)

Программный код, решающий задачу:

```
p = 30000
```

```
t = 5
```

```
s = 40000
```

```
j12 = (12*((s/p)**(1/(12 *t))) - 12) * 100
```

```
print(round(j12,2))
```

Скриншот с ответом:

The screenshot shows a Python Shell interface. The code is entered line by line, and the output is displayed at the bottom right. The code is as follows:

```
1 p = 30000
2 t = 5
3 s = 40000
4 j12 = (12*((s/p)**(1/(12 *t))) - 12) * 100
5 print(round(j12,2))
```

The output of the program is 5.77, which is displayed in the 'Debug I/O' panel at the bottom right of the interface.

Задача 2.

Банк выплачивает по вкладам 8% годовых (сложных). Какова реальная доходность вкладов в этот банк, если начисление поквартально?

В данной задаче достаточно применить формулу эквивалентности ставки сложных процентов и годовой ставке j_m , по которой m раз в год начисляется j_m/m сложных процентов. Формула (3.20)

$$i_c = \left(1 + \frac{j_m}{m}\right)^m - 1 \quad (3.20)$$

Программный код, решающий задачу:

```
j4 = 8
```

```
ic = ((1 + j4 / 400)**4 - 1)*100
```

```
print(round(ic,2))
```

Скриншот с ответом:

```
1 j4 = 8
2 ic = ((1 + j4 / 400)**4 - 1)*100
3 print(round(ic,2))
```



Задача 3.

Выведите формулу и сформулируйте правило, аналогичное правилу 69, для определения срока, за который сумма, размещенная под r сложных процентов за период, утроится. За какой срок по этому правилу сумма, размещенная под 25% сложных процентов за период, утроится? Сравните найденное значение срока с точным, вычисленным по формуле (3.34). Как бы вы назвали это правило?

Вопрос 1 – вывод формулы изображен ниже в рис.1. Вопросы 2 и 3 – представлены в программном коде и в скриншоте с ответом.

Вывод формулы: $S = P(1+r)^t$
для утроения $S = 3P$
 $(1+r)^t = 3$
 $t = \frac{\ln 3}{\ln(1+r)}$ $\ln 3 = 1,0986 \approx 1,1$
 $\ln(1+r) = r - \frac{r^2}{2}$ (разложение в ряд Маклорена), другие слагаемые не записываю, т.к. при $-1 < r \leq 1$ они будут $O(r^2)$
Тогда $t \approx \frac{1}{r - \frac{r^2}{2}} = \frac{1}{r(1 - \frac{r}{2})}$ $\frac{1}{1 - \frac{r}{2}} = \sum_{k=0}^{\infty} (\frac{r}{2})^k \approx 1 + \frac{r}{2}$ (разложение в ряд)
 $t \approx \frac{1,1}{r} (1 + \frac{r}{2}) = \frac{1,1}{r} + 0,55 = \frac{110}{r \cdot 100} + 0,55 \Rightarrow t = \frac{110}{r} + 0,55$
в дальнейшем преобразовании заменил r , подразумевая, что если в задаче дано $r = 30\%$, то в формулу подставлю $r = 30$.
ответ: $t = \frac{110}{r} + 0,55$

Рис. 1

Вопрос 4 – “Правило 110”.

Программный код, решающий задачу:

```
r = 25
```

```
t3 = 110/r + 0.55
```

```
t2 = 69 / r + 0.35
```

```
print(round(t3,2))
```

```
if t3 > t2:
```

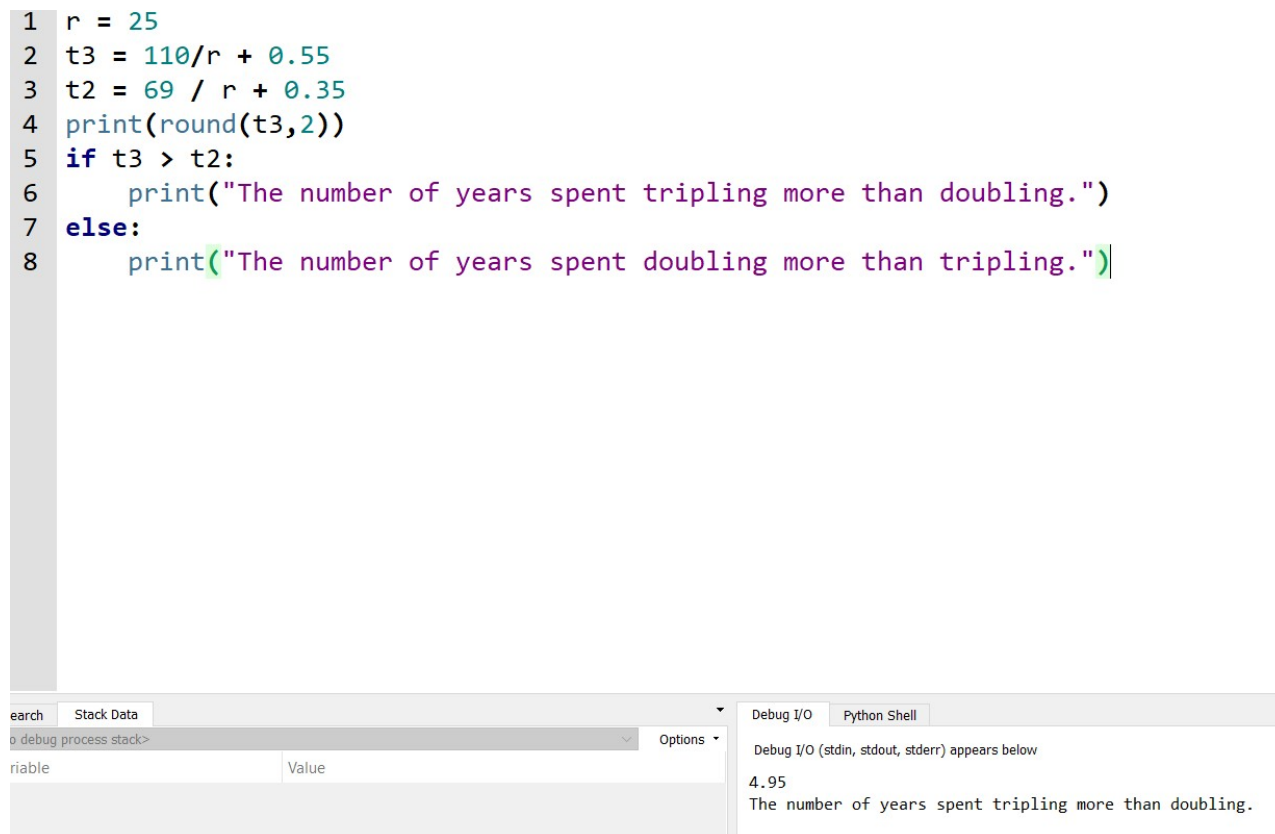
```
    print("The number of years spent tripling more than doubling.")
```

```
else:
```

```
    print("The number of years spent doubling more than tripling.")
```


Скриншот с ответом:

```
1 r = 25
2 t3 = 110/r + 0.55
3 t2 = 69 / r + 0.35
4 print(round(t3,2))
5 if t3 > t2:
6     print("The number of years spent tripling more than doubling.")
7 else:
8     print("The number of years spent doubling more than tripling.")
```



Variable	Value
t3	4.95

Debug I/O (stdin, stdout, stderr) appears below

4.95
The number of years spent tripling more than doubling.

Заключение

В заключение хотелось бы сказать, что цель, которая была поставлена передо мной, была достигнута. Сейчас мне намного легче работать с процентами, я наконец-то начал понимать, как работают кредиты и вклады в банках, изучил такие виды ценных бумаг, как акции, облигации, вексели. Также хочу отметить, что были полезны примеры, основанные на реальных жизненных ситуациях. Благодаря всей этой информации у меня появилась минимальная база, которая поможет мне в будущем грамотнее управлять моими денежными средствами.

Список используемой литературы

1. Бухвалов А.В., Бухвалова В.В. Финансовые вычисления для менеджеров. – СПб.: Изд-во «Высшая школа менеджмента», 2010.
2. Swaroop С.Н. A byte of Python. <https://python.swaroopch.com> (17.12.2020)