**Поволжский Государственный Университет Телекоммуникаций и Информатики**

**Кафедра ПОУТС**

Отчет по лабораторной работе №4

«Определение и анализ рыночной стоимости прикладного программного обеспечения»

Вариант №6

Выполнили: студенты группы ПО-61

Булычев Иван

Проверила: Вержаковская М. А.

Самара 2020

**Цель работы** – освоить методику определения и анализа рыночной стоимости прикладного программного обеспечения.

**Порядок выполнения работы:**

1. Определить точку безубыточности, т.е. минимальное количество (объем продаж) по заданной договорной цене, при котором выручка покроет затраты предприятия на его создание и тиражирование.

Построить график точки безубыточности.

1. Рассчитать договорную цену тиражируемого программного продукта при заданном объеме рынка продаж.
2. Определить объем продаж при заданном уровне прибыли и рыночной цене тиражируемого программного продукта.
3. Определить срок окупаемости проекта при заданной договорной цене тиражируемого программного продукта в точке безубыточности и общее количество копий для полного возмещения затрат.

**Результат работы**

**Вариант 6.**

Содержимое файла **params.properties** с исходными данными для расчета:

договорная\_цена(стоимость\_разработки\_ПС\_из\_3лб)=50252240.59

срок\_месяцев=8

процент\_банковского\_кредита=0.16

заданный\_обьем\_рынка\_продаж=25.0

дополнительная\_прибыль=250000.0

зарплата\_специалистов\_отдела\_маркетинга\_проценты=0.5

**Результат работы программы на Python:**

Лабораторная работа №4

Определение и анализ рыночной стоимости прикладного программного обеспечения

Стоимость продажи одной копии системы: 2512612.03

Таблица 2.1 Постоянные(фиксированные) расходы в месяц

Наименование расходов Сумма (руб)

------------------------------------------------------------ -------------

Накладные расходы по проекту - 10% от ФЗП АУП 100000.00

Плановое ежемесячное гашение кредита 6281530.07

Выплата среднего банковского процента 670029.87

Прочие расходы - 10% от накладных расходов на содержание АУП 10000.00

ИТОГО 7061559.95

Таблица 2.2 – Переменные издержки(отдел маркетинга)

Наименование расходов Сумма (руб)

------------------------------------------------------------------------------ -------------

Основная зарплата специалистов 0.5 процент от стоимости тиражируемого продукта 1256306.01

Страховые взносы (30%) от фонда зарплаты 30000.00

Комплектующие и расходные материалы 1% от стоимости тиражируемого продукта 25126.12

Накладные расходы отдела маркетинга 1.5% от стоимости тиражируемого продукта% 37689.18

ИТОГО: 1349121.82

Объем выпуска, при котором достигается точка безубыточности (нулевой уровень прибыли), определяется по формуле:

x0 = a / (s - b) = 7

Выводы

В течение месяца фирме необходимо подготовить и продать минимум 7 копий программного продукта по цене 2512612.03 чтобы окупить постоянные и переменные расходы.

2.2 Расчет договорной цены тиражируемой системы при заданном объеме рынка продаж:

Sm = 1631584.21

Скидка оптовому покупателю: 35.06 %

2.3 Определение дополнительного объема продаж при заданном уровне прибыли

Xd = 7

Выводы. Объем продаж для получения дополнительной прибыли в размере 250000.0 рублей составляет 6.284161104430898 копий продукта в месяц, при условии, что постоянные и переменные издержки фирмы неизменны

2.4 Определение срока окупаемости проекта и количества продаж для полного возмещения затрат

Xn = 48.55432293791492

Cok = 8.0

Выводы. срок окупаемости проекта при продаже не менее 7 копий продукта в месяц(точка безубыточности) и рыночной стоимости 1631584.21 руб. за копию составит 8.0 месяцев. Для того, чтобы окупить все расходы на реализацию проекта, необходимо продать 49 копий программного продукта

Press ENTER to exit

**Графический анализ точки безубыточности:**

A close up of a map

Description automatically generated

**Приложение**

Актуальные версии исходного кода и исполняемых файлов можно посмотреть по [ссылке](https://github.com/MaximKozlyuk/SoftwareEconomicsPSUTI).

Листинг программы lab4.py

import math

import matplotlib.pyplot as plt

import os

import sys

from tabulate import tabulate

class DefaultPath(object):

def \_\_init\_\_(self, file\_name) -> None:

if getattr(sys, 'frozen', False):

# If the application is run as a bundle, the pyInstaller bootloader

# extends the sys module by a flag frozen=True and sets the app

# path into variable \_MEIPASS'.

self.app\_path = sys.\_MEIPASS

else:

self.app\_path = os.path.dirname(os.path.abspath(\_\_file\_\_))

self.app\_path = os.path.dirname(sys.argv[0])

self.default\_properties\_path = self.app\_path + os.path.sep + "params.properties"

if len(file\_name) != 0 and file\_name != "":

self.file\_name = self.app\_path + os.path.sep + file\_name

super().\_\_init\_\_()

class PropertiesFile(DefaultPath):

def \_\_init\_\_(self, file\_name) -> None:

super().\_\_init\_\_(file\_name)

def properties(self, delimiter):

p = []

with open(self.file\_name, 'r') as file:

for row in file:

parts = row.split(str(delimiter))

try:

p.append((parts[0], float(parts[1])))

except ValueError as e: # to remove new line symbol in case if reading string value

p.append((parts[0], str(parts[1])[:len(str(parts[1]))-1]))

return p

if getattr(sys, 'frozen', False):

# If the application is run as a bundle, the pyInstaller bootloader

# extends the sys module by a flag frozen=True and sets the app

# path into variable \_MEIPASS'.

application\_path = sys.\_MEIPASS

else:

application\_path = os.path.dirname(os.path.abspath(\_\_file\_\_))

application\_path = os.path.dirname(sys.argv[0])

def round\_money(x):

return round(x, 2)

def print\_money(msg, x):

print(msg, round\_money(x))

def calc\_fixed\_spends(arr):

s = sum(arr)

rows = []

headers = ["Наименование расходов", "Сумма (руб)"]

rows.append(headers)

rows.append(["Накладные расходы по проекту - 10% от ФЗП АУП", arr[0]])

rows.append(["Плановое ежемесячное гашение кредита", arr[1]])

rows.append(["Выплата среднего банковского процента", arr[2]])

rows.append(["Прочие расходы - 10% от накладных расходов на содержание АУП", arr[3]])

rows.append(["ИТОГО", s])

print(tabulate(rows, headers="firstrow", floatfmt=".2f"), "\n")

return s

def calc\_var\_spends(arr):

s = sum(arr)

rows = []

headers = ["Наименование расходов", "Сумма (руб)"]

rows.append(headers)

zp = "Основная зарплата специалистов %s процент от стоимости тиражируемого продукта" % (round\_money(arr[0]))

rows.append([zp, arr[1]])

rows.append(["Страховые взносы (30%) от фонда зарплаты", arr[2]])

rows.append(["Комплектующие и расходные материалы 1% от стоимости тиражируемого продукта", arr[3]])

rows.append(["Накладные расходы отдела маркетинга 1.5% от стоимости тиражируемого продукта% ", arr[4]])

rows.append(["ИТОГО:", s])

print(tabulate(rows, headers="firstrow", floatfmt=".2f"), "\n")

return s

props = PropertiesFile("params.properties")

props = props.properties("=")

# Стоимость программной системы из лб 3

PS\_cost = props[0][1]

# Срок месяцев

deadline = props[1][1]

# Процент банковского кредита

bank\_credit\_percent = props[2][1]

# Заданный объем рынка продаж

market\_v = props[3][1]

# Дополнительная прибыль (НЕ тыс. рублей а просто рублей)

additional\_profit = props[4][1]

# ЗП специалистов отдела маркетинга (%)

market\_stuff\_salary\_percent = props[5][1]

print("Лабораторная работа №4\nОпределение и анализ рыночной стоимости прикладного программного обеспечения\n")

one\_copy\_cost = PS\_cost \* 0.05 # стоимость тиражируемого продукта

print\_money("Стоимость продажи одной копии системы:", one\_copy\_cost)

marketing\_employ\_salary = one\_copy\_cost \* market\_stuff\_salary\_percent

print("Таблица 2.1 Постоянные(фиксированные) расходы в месяц")

managers\_overhead\_spends = 50000.0 + 30000.0 + 20000.0 # накладные расходы на содержание АУП

month\_credit\_deposit = PS\_cost / deadline

month\_avg\_credit\_proc = (PS\_cost \* bank\_credit\_percent) / 12

other\_spends = managers\_overhead\_spends \* 0.1

fixed\_spends = calc\_fixed\_spends([managers\_overhead\_spends, month\_credit\_deposit, month\_avg\_credit\_proc, other\_spends])

print("Таблица 2.2 – Переменные издержки(отдел маркетинга)")

insurance\_premiums = managers\_overhead\_spends \* 0.3

comp\_expendable\_materials = one\_copy\_cost \* 0.01

comp\_expendable\_materials\_marketing = one\_copy\_cost \* 0.015

var\_spends = calc\_var\_spends([

market\_stuff\_salary\_percent,

marketing\_employ\_salary,

insurance\_premiums,

comp\_expendable\_materials,

comp\_expendable\_materials\_marketing

])

v\_zero\_point = fixed\_spends / (one\_copy\_cost - var\_spends)

print("Объем выпуска, при котором достигается точка безубыточности (нулевой уровень прибыли), определяется по формуле:")

print("x0 = a / (s - b) =", math.ceil(v\_zero\_point))

print("\nВыводы")

print("В течение месяца фирме необходимо подготовить и продать минимум", math.ceil(v\_zero\_point), "копий программного"

, "продукта по цене", round\_money(one\_copy\_cost), "чтобы окупить постоянные и переменные расходы.")

# a - fixed\_spends, b - var\_spends, xp - market\_v

# расчет значений для графика, на 20% больше чем точка безубыточности

ceil\_zero\_point = math.ceil(v\_zero\_point)

sales\_range = ceil\_zero\_point + (math.ceil(ceil\_zero\_point \* 0.2)) + 1

# точки x для всех графиков одинаковы (кол-во рподанных товаров)

x\_points = list(range(0, sales\_range))

var\_spends\_counter = 0.0

var\_spends\_y\_points = []

sales\_income\_counter = 0.0

sales\_income\_y\_points = []

total\_costs\_counter = fixed\_spends

total\_costs\_y\_points = []

for i in range(0, sales\_range):

var\_spends\_y\_points.append(var\_spends\_counter)

sales\_income\_y\_points.append(sales\_income\_counter)

sales\_income\_counter += one\_copy\_cost

total\_costs\_y\_points.append(total\_costs\_counter)

total\_costs\_counter += var\_spends

var\_spends\_counter += var\_spends

plt.plot([0, sales\_range - 1], [fixed\_spends, fixed\_spends], label="Постоянные издержки", color="green")

plt.plot(x\_points, var\_spends\_y\_points, label="Перменные издержки", color="brown", linestyle="dashed")

plt.plot(x\_points, sales\_income\_y\_points, label="Доходы от продаж", color="blue", linestyle="dashed")

plt.plot(x\_points, total\_costs\_y\_points, label="Общие издержки", color="brown")

zero\_point\_income = one\_copy\_cost \* v\_zero\_point

plt.plot(

[v\_zero\_point, v\_zero\_point], [0, zero\_point\_income],

label="Точка безубыточности", color="red", linestyle="dashed"

)

plt.plot([v\_zero\_point], [zero\_point\_income], color="red", marker="o")

plt.xlabel("Количество продаж")

plt.ylabel("₽")

plt.grid()

plt.legend()

plt.show()

print("2.2 Расчет договорной цены тиражируемой системы при заданном объеме рынка продаж:")

Sm = (fixed\_spends + var\_spends \* market\_v) / market\_v

wholesale\_discount = 100 - (Sm \* 100 / one\_copy\_cost)

print("Sm =", round\_money(Sm))

print("Скидка оптовому покупателю:", round\_money(wholesale\_discount), "%")

print("2.3 Определение дополнительного объема продаж при заданном уровне прибыли")

Xd = (additional\_profit + fixed\_spends) / (one\_copy\_cost - var\_spends)

print("Xd =", math.ceil(Xd))

print(

"Выводы. Объем продаж для получения дополнительной прибыли в размере", additional\_profit,

"рублей составляет", Xd,

"копий продукта в месяц, при условии, что постоянные и переменные издержки фирмы неизменны"

)

print("2.4 Определение срока окупаемости проекта и количества продаж для полного возмещения затрат")

# todo по тексту методички не ясно, мб тут вместо one\_copy\_cost Sm (договорная стоимость продажи системы)

Xn = (deadline \* fixed\_spends) / (one\_copy\_cost - var\_spends)

Cok = Xn / v\_zero\_point

print("Xn =", Xn)

print("Cok =", Cok)

print(

"Выводы. срок окупаемости проекта при продаже не менее", ceil\_zero\_point, "копий продукта в месяц"

"(точка безубыточности) и рыночной стоимости", round\_money(Sm), "руб. за копию составит", round(Cok, 2), "месяцев. "

"Для того, чтобы окупить все расходы на реализацию проекта, необходимо продать",

math.ceil(Xn), "копий программного продукта"

)

input('\nPress ENTER to exit')

exit(0)