


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
Национальный исследовательский университет ИТМО
ФАКУЛЬТЕТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Управление мобильными устройствами
Лабораторная работа №3
«Формирование счета на оплату услуг»
Вариант №11

Работу выполнил:

Студент группы №3352
Ласкус Е.О.


23.05.20

Работу проверил:

Федоров И. Р . _____



Санкт-Петербург, 2020

Цель работы: разработка и реализация программного модуля создания и автоматического формирования счета на оплату по данным, полученным в предыдущих работах.

Описание выбранных средств реализации:

Разработанный программный модуль был реализован на языке Python с использованием библиотеки обработки и анализа данных Pandas.

Одна из причин выбора Python— более простое оформление, чем в других языках. Его проще изучить с нуля даже новичку в программировании. Это самый популярный язык общего назначения: он используется для машинного обучения, аналитике, разработке игр и в науке о данных.

За основу для создания PDF файла была взята библиотека PyFPDF, так как она позволяет собирать PDF файл из структур, схожих со структурами html.

Исходный код:

MLW_3.py

```
# -*- coding: utf-8 -*-
"""Copy of MLW_3.ipynb

Automatically generated by Colaboratory.

Original file is located at
https://colab.research.google.com/drive/13VklgHfV6GbCeIK5hWqDS71dLMUyae4s
"""

from fpdf import FPDF
import datetime
import math
import os
import sys

def banks (pdf: FPDF, **kwargs):
    MAXIMUM_PAGE_WIDTH = 210
    PADDING = 20
    height = 30
    l_col_w = 95
    m_col_w = 16
    r_col_w = MAXIMUM_PAGE_WIDTH - PADDING * 2 - l_col_w - m_col_w

    pdf.line(PADDING, PADDING, MAXIMUM_PAGE_WIDTH - 20, PADDING)
    pdf.line(PADDING, PADDING, PADDING, height + PADDING)
    pdf.line(MAXIMUM_PAGE_WIDTH - PADDING, PADDING, MAXIMUM_PAGE_WIDTH - PADDING, height +
PADDING)
    pdf.line(PADDING, height + PADDING, MAXIMUM_PAGE_WIDTH - PADDING, height + PADDING)
    pdf.line(PADDING, height * 9 / 21 + PADDING, MAXIMUM_PAGE_WIDTH - PADDING, height * 9 / 21 + PADDING)
    pdf.line(PADDING, height * 12 / 21 + PADDING, l_col_w + PADDING, height * 12 / 21 + PADDING)
    pdf.line(l_col_w + PADDING, PADDING, l_col_w + PADDING, height + PADDING)
    pdf.line(l_col_w + PADDING + m_col_w, PADDING, l_col_w + PADDING + m_col_w, height + PADDING)
    pdf.line(l_col_w + PADDING, PADDING + height * 3 / 21, l_col_w + m_col_w + PADDING, PADDING + height * 3 /
21)
    pdf.line(PADDING + l_col_w / 2, PADDING + height * 9 / 21, PADDING + l_col_w / 2, PADDING + height * 12 / 21)
    pdf.line(PADDING, height + 15, MAXIMUM_PAGE_WIDTH / 1.81 - 1, height + 15)
    pdf.set_font("TimesNewRoman", size=9)
    pdf.set_y(PADDING)
    pdf.cell(10)
    pdf.multi_cell(95, 4, kwargs['payee_bank'])
    pdf.set_y(PADDING + 12.5)
    pdf.cell(10)
    pdf.cell(l_col_w / 2, 5, f'ИНН {kwargs["INN"]}')
    pdf.cell(l_col_w / 2, 5, f'КПП {kwargs["KPP"]}')
    pdf.set_y(PADDING + 17.5)
    pdf.cell(10)
    pdf.multi_cell(95, 4, kwargs['payee'])
    pdf.set_y(PADDING)
    pdf.cell(10 + l_col_w)
    pdf.cell(m_col_w, 5, 'БИК')
    pdf.cell(r_col_w, 5, kwargs['BIK'])
```

```

pdf.set_y(PADDING + 5)
pdf.cell(10 + l_col_w)
pdf.cell(m_col_w, 5, 'Сч. №')
pdf.cell(r_col_w, 5, kwargs['account1'])
pdf.set_y(PADDING + height * 9 / 21)
pdf.cell(10 + l_col_w)
pdf.cell(m_col_w, 5, 'Сч. №')
pdf.cell(r_col_w, 5, kwargs['account2'])
pdf.set_font("TimesNewRoman", size=8)
pdf.set_y(PADDING + height * 6 / 21)
pdf.cell(10)
pdf.cell(l_col_w, 4, 'Банк получателя')
pdf.set_y(PADDING + height * 18 / 21)
pdf.cell(10)
pdf.cell(l_col_w / 2, 5, f'Получатель {kwargs["addressee"]}')

```

```

return PADDING + height

```

```

def payment_invoice(pdf: FPDF, **kwargs):
    MAXIMUM_PAGE_WIDTH = 210
    PADDING = 20
    pdf.set_font("TimesNewRomanB", size=10)
    pdf.set_y(kwargs['height'] + 4)
    pdf.cell(10)
    pdf.cell(MAXIMUM_PAGE_WIDTH - PADDING, 9, f'Счет на оплату № {kwargs["account_number"]} от {kwargs["day"]}.{kwargs["month"]}.20{kwargs["year"]} г.')

```

```

return kwargs['height'] + 14.1

```

```

def requisites(pdf: FPDF, **kwargs):
    MAXIMUM_PAGE_WIDTH = 210
    PADDING = 20
    l_col_w = 28
    r_col_w = MAXIMUM_PAGE_WIDTH - 2 * PADDING - l_col_w
    line_height = 5
    pdf.set_font("TimesNewRoman", size=9)
    pdf.set_y(kwargs['height'] + 2)
    pdf.cell(10)
    pdf.multi_cell(l_col_w, line_height, 'Поставщик:')
    pdf.set_font("TimesNewRomanB", size=9)
    pdf.set_y(kwargs['height'] + 2)
    pdf.cell(10 + l_col_w)
    executor = pdf.multi_cell(r_col_w, line_height, kwargs['executor'], split_only=True)
    pdf.multi_cell(r_col_w, line_height, kwargs['executor'])
    height = kwargs['height'] + 6 + 5 * len(executor)
    pdf.set_font("TimesNewRoman", size=9)
    pdf.set_y(height)
    pdf.cell(10)
    pdf.multi_cell(l_col_w, line_height, 'Покупатель:')
    pdf.set_font("TimesNewRomanB", size=9)
    pdf.set_y(height)
    pdf.cell(10 + l_col_w)
    buyer = pdf.multi_cell(r_col_w, line_height, kwargs['client'], split_only=True)
    pdf.multi_cell(r_col_w, line_height, kwargs['client'])
    height += len(buyer) * 5 + 6
    pdf.set_y(height)
    pdf.cell(10)

```

```

pdf.set_font("TimesNewRoman", size=9)
pdf.cell(l_col_w, line_height, 'Основание:')
pdf.set_font("TimesNewRomanB", size=9)
pdf.cell(r_col_w, line_height, kwargs['cause'])

return height + line_height

def services(pdf: FPDF, goods, **kwargs):
    MAXIMUM_PAGE_WIDTH = 210
    PADDING = 20
    col1_w = 8
    col3_w = 20
    col4_w = 18
    col5_w = 18
    col2_w = MAXIMUM_PAGE_WIDTH - 2 * PADDING - col1_w - col3_w - col4_w - col5_w
    height = kwargs['height'] + 6
    start = height
    total = 0
    pdf.set_line_width(0.2)
    pdf.line(PADDING, height, MAXIMUM_PAGE_WIDTH - PADDING, height)
    pdf.set_line_width(0.2)
    pdf.line(PADDING, height + 5, MAXIMUM_PAGE_WIDTH - PADDING, height + 5)
    pdf.set_font("TimesNewRomanB", size=9)
    pdf.set_y(height)
    pdf.cell(10)
    pdf.cell(col1_w, 5, '№', align='C')
    pdf.cell(col2_w, 5, 'Товары (работы, услуги)', align='C')
    pdf.cell(col3_w, 5, 'Кол-во', align='C')
    pdf.cell(col4_w, 5, 'Цена', align='C')
    pdf.cell(col5_w, 5, 'Сумма', align='C')

    pdf.set_font("TimesNewRoman", size=9)
    height += 6
    index = 0
    for id, good in enumerate(goods):
        is_last = id == len(goods) - 1
        index += 1
        pdf.set_y(height)
        pdf.cell(10)
        pdf.cell(col1_w, 5, str(index), align='C')
        pdf.multi_cell(col2_w, 5, good['service'])
        pdf.set_y(height)
        pdf.cell(10 + col1_w + col2_w)
        pdf.cell(col3_w, 5, f'{math.ceil(good["total"])}{" " + good["total_unit"] if "total_unit" in good else ""}', align='L')
        pdf.cell(col4_w, 5, f'{math.ceil(good["coefficient"])}{"p/" + good["total_unit"] if "total_unit" in good else "p"}', align='L')
        if good['service'] == 'CMC':
            pdf.cell(col5_w, 5, f'{math.ceil(good["coefficient"]) * math.ceil(good["total"]) - 5} p', align='R')
            total += math.ceil(good["coefficient"]) * math.ceil(good["total"]) - 5
            height += 5 * len(pdf.multi_cell(col2_w, 100, good['service'], split_only=True))
            pdf.line(PADDING, height + 1, MAXIMUM_PAGE_WIDTH - PADDING, height + 1)
            height += 2
            continue

        pdf.cell(col5_w, 5, f'{math.ceil(good["coefficient"]) * math.ceil(good["total"])} p', align='R')
        total += math.ceil(good["coefficient"]) * math.ceil(good["total"])
        height += 5 * len(pdf.multi_cell(col2_w, 100, good['service'], split_only=True))

    if not is_last:

```

```

pdf.line(PADDING, height + 1, MAXIMUM_PAGE_WIDTH - PADDING, height + 1)
height += 2
else:
    height += 1

pdf.set_line_width(0.2)
pdf.line(PADDING, start, PADDING, height)
pdf.line(PADDING, height, MAXIMUM_PAGE_WIDTH - PADDING, height)
pdf.line(MAXIMUM_PAGE_WIDTH - PADDING, start, MAXIMUM_PAGE_WIDTH - PADDING, height)
pdf.set_line_width(0.2)
pdf.line(PADDING + col1_w, start, PADDING + col1_w, height)
pdf.line(PADDING + col1_w + col2_w, start, PADDING + col1_w + col2_w, height)
pdf.line(PADDING + col1_w + col2_w + col3_w, start, PADDING + col1_w + col2_w + col3_w, height)
pdf.line(PADDING + col1_w + col2_w + col3_w + col4_w, start, PADDING + col1_w + col2_w + col3_w + col4_w,
height)
pdf.line(PADDING + col1_w + col2_w + col3_w + col4_w + col5_w, start, PADDING + col1_w + col2_w + col3_w +
col4_w + col5_w, height)

height += 5
pdf.set_font("TimesNewRomanB", size=9)
pdf.set_y(height)
total_price = f'{total:,.2f}'.replace(',', ' ')
NDS = f'{total * 0.167:,.2f}'.replace(',', ' ')
pdf.cell(10)
pdf.multi_cell(10 + MAXIMUM_PAGE_WIDTH - 2 * PADDING, 5, f'Итого: {total_price:>15} p.', align='L')
pdf.cell(10)
pdf.multi_cell(10 + MAXIMUM_PAGE_WIDTH - 2 * PADDING, 5, f'В том числе НДС: {NDS:>15} p.', align='L')
pdf.cell(10)
pdf.multi_cell(10 + MAXIMUM_PAGE_WIDTH - 2 * PADDING, 5, f'Всего к оплате: {total_price:>15} p.', align='L')

pdf.set_font("TimesNewRoman", size=9)
pdf.cell(10)
pdf.multi_cell(MAXIMUM_PAGE_WIDTH - 2 * PADDING, 5, f'Всего наименований {index} на сумму {total_price}
руб.')
pdf.set_font("TimesNewRomanB", size=9)

return height + 25

def conditions(pdf: FPDF, **kwargs):
    def add_text(text, height):
        MAXIMUM_PAGE_WIDTH = 210
        PADDING = 20
        pdf.cell(10)
        height += 4 * len(pdf.multi_cell(MAXIMUM_PAGE_WIDTH - 2 * PADDING, 4, text, split_only=True))
        pdf.multi_cell(MAXIMUM_PAGE_WIDTH - 2 * PADDING, 4, text)
        return height
    height = kwargs['height'] + 10
    pdf.set_font("TimesNewRoman", size=8)
    pdf.set_y(height)
    height = add_text('Внимание!', height)
    height = add_text('Оплата данного счета означает согласие с условиями поставки товара.', height)
    height = add_text('Уведомление об оплате обязательно, в противном случае не гарантируется наличие
товара на складе.', height)
    height = add_text('Товар отпускается по факту прихода денег на р/с Поставщика, самовывозом, при наличии
доверенности и паспорта.', height)

    MAXIMUM_PAGE_WIDTH = 210
    PADDING = 20
    pdf.set_y(height)

```

```

pdf.set_font("TimesNewRomanB", size=9)
pdf.cell(10)
pdf.cell(30, 5, 'Руководитель')
pdf.set_font("TimesNewRoman", size=9)
pdf.cell(60, 5, kwargs['supervisor'], align='R')
pdf.set_font("TimesNewRomanB", size=9)
pdf.cell(30, 5, 'Бухгалтер', align='C')
pdf.set_font("TimesNewRoman", size=9)
pdf.cell(MAXIMUM_PAGE_WIDTH - 2 * PADDING - 120, 5, kwargs['accountant'], align='R')

pdf.set_line_width(0.2)
pdf.line(PADDING + 35, height + 5, PADDING + 90, height + 5)
pdf.line(PADDING + 120, height + 5, MAXIMUM_PAGE_WIDTH - PADDING, height + 5)

return height

pdf = FPDF()
pdf.add_page()
pdf.add_font('TimesNewRoman', '', 'fonts/TNR.ttf', uni=True)
pdf.add_font('TimesNewRomanB', '', 'fonts/TNRB.ttf', uni=True)
now = datetime.datetime.today()

height = banks(pdf,
    payee_bank = 'ООО "Айрон Банк"',
    INN = '7722737766',
    KPP = '772201001',
    payee = 'ООО "Харвестер"',
    BIK = '044525700',
    account1 = '30101810200000000700',
    account2 = '407028109000000002453',
    addressee = 'ООО "Шай-Хулуд"')

height = payment_invoice(pdf,
    account_number = 42,
    day = now.day,
    month = '{:02d}'.format(now.month),
    year = 20,
    height = height)

height = requisites(pdf,
    height=height,
    executor='ООО "Смерть за Долги", '
        'Вольные Города, г. Браавос, домовладение 10, стр.1',
    client='ООО "Шай-Хулуд", ИНН 7714037378, КПП 777550001, 119361, '
        'Аппакис, равнина Лето II, плато Гейрат',
    cause='№ 11223344 от 01.01.2019')

height = services(pdf, [
    {
        'service': 'Входящие вызовы',
        'total': 9.2,
        'total_unit': 'мин.',
        'coefficient': 1,
    }, {
        'service': 'Исходящие вызовы',
        'total': 36.23,
        'total_unit': 'мин.',
        'coefficient': 1,
    }, {

```

```
'service': 'СМС',
'total': 5,
'total_unit': 'шт.',
'coefficient': 1,
}, {
'service': 'Исходящий трафик',
'total': 0.0392,
'total_unit': 'Мб',
'coefficient': 0.5,
}, {
'service': 'Входящий трафик',
'total': 0.0108,
'total_unit': 'Мб',
'coefficient': 0.5,
},
], height=height)

height = conditions(pdf,
                    height=height,
                    supervisor='Семенов Д.А',
                    accountant='Семенов Д.А.')

pdf.output('Payment.pdf')
```

Вывод:

В данной Лабораторной работе были изучены принципы и технологии разработки и реализации программного модуля создания и автоматического формирования счета на оплату по данным, полученным в предыдущих работах.