Министерство науки и высшего образования РФ

ФГАОУ ВО ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИУ)

Институт естественных и точных наук

Факультет математики, механики и компьютерных технологий

Кафедра прикладной математики и программирования

«Программа для оптимизации работы станков с ЧПУ»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОЕКТУ   
по дисциплине «Учебная практика»

ЮУрГУ–01.03.02.2022.112.ПЗ

|  |  |  |
| --- | --- | --- |
|  |  | *Руководитель,*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Демидов А.К.*  *«* *»*   *2022г.* |
|  |  | *Авторы работы:*  *Студент группы: ЕТ – 112*  *Никитин М.С.*  *Студент группы: ЕТ – 112*  *Мухутдинов Б.А.*  *Студент группы: ЕТ – 112*  *Дерина В.Д.*  *Студент группы: ЕТ – 112*  *Забалдина Д.И.*  *Студент группы: ЕТ – 112*  *Дюрягина К.Ю.*  *Студент группы: ЕТ – 112*  *Шафикова М.А.*  *«* *»*   *2022г.* |
|  |  | *Работа защищена с оценкой*    *«* *»*   *2022г.* |

Челябинск – 2022ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ](#__RefHeading___Toc4863_2799259349) 3

[1 ПОСТАНОВКА ЗАДАЧИ](#__RefHeading___Toc7484_2799259349) 4

[2 РАЗРАБОТКА АЛГОРИТМА](#__RefHeading___Toc7486_2799259349) 5

[3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ](#__RefHeading___Toc7272_2799259349) 8

[ЗАКЛЮЧЕНИЕ](#__RefHeading___Toc7164_2799259349) 10

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК](#__RefHeading___Toc7166_2799259349) 11

[ПРИЛОЖЕНИЕ а](#__RefHeading___Toc7499_2626037720) 12

# ВВЕДЕНИЕ

**Актуальность темы**. Операторы станков с ЧПУ (Числового Программного Управления) используют низкие скорости при производстве, потому что боятся брака деталей.

**Цель работы** – разработать программное обеспечение для операторов станков с ЧПУ для демонстрации оптимальных режимов работы.

**Задачи работы**:

– изучить работу станков ЧПУ;

– изучить основные параметры, которые необходимы для программирования станков ЧПУ;

– составить макет и алгоритм программы, согласовать интерфейс;

– провести тестирование.

**Объект работы** – программа для оптимизации работы станков с ЧПУ.

**Предмет работы** – применение технологий коллективной разработки программного обеспечения для разработки программы.

**Результаты работы** можно использовать для дальнейшего нахождения наиболее оптимальных методов оптимизации работы станков с ЧПУ.

## 1 ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать программу для вычисления и демонстрации оптимального режима работы для станка ЧПУ на основе введенных начальных данных.

Перечень основных групп данных включает в себя

|  |  |
| --- | --- |
| Номер | Основные группы данных |
| 1 | По шлифовальному кругу |
| 2 | По детали |
| 3 | По заготовке |
| 4 | По станку |
| 5 | По циклу |

Построение графика по введенным числами осуществляется по нажатию специальной кнопки с названием «Результаты».

Скачивание и загрузка данных для построения графика происходят с помощью кнопок «Экспорт» и «Импорт».

Для разработки был использован язык программирования Python и графическая библиотека Qt5.

Интерфейс начального окна представлен на рисунке 1, интерфейс окна ввода параметров – на рисунке 2.

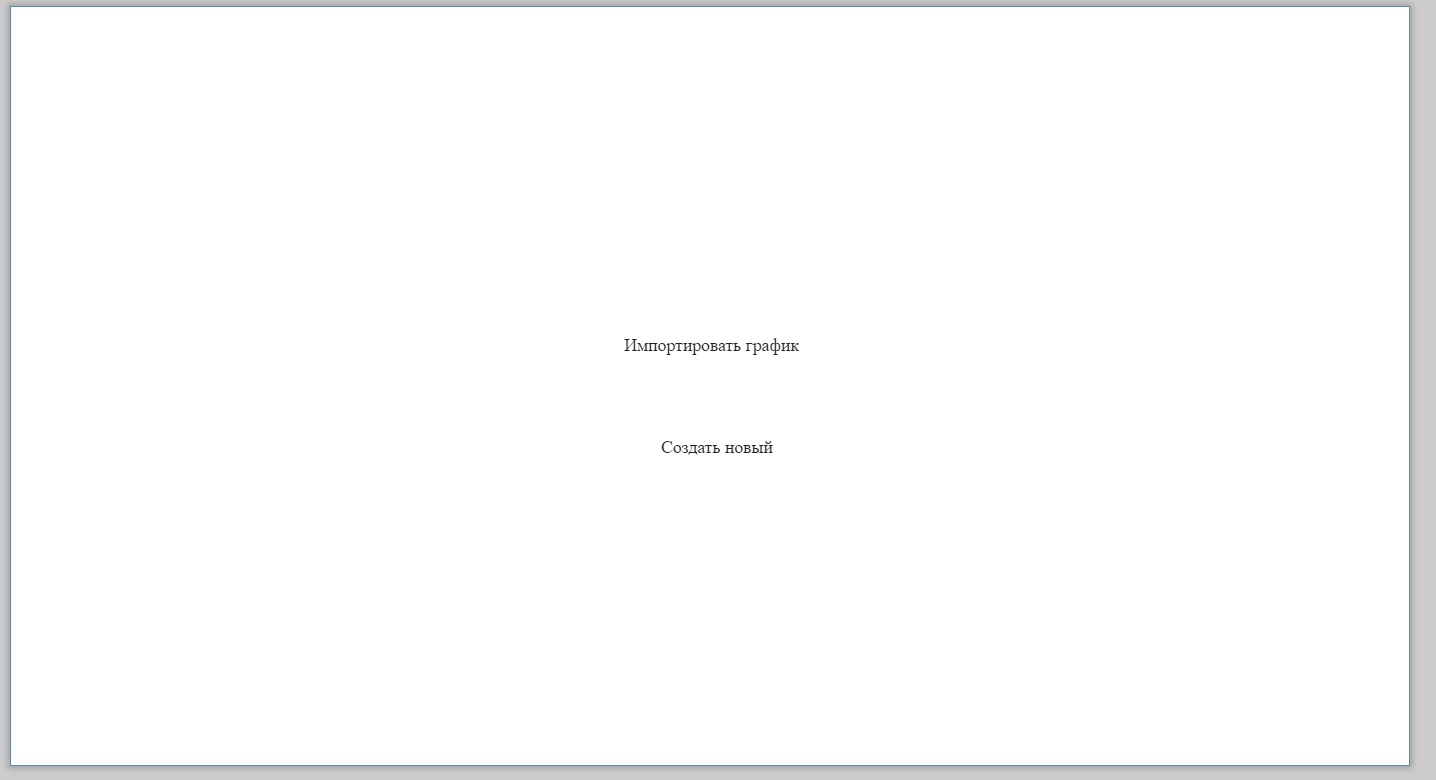


Рисунок 1 - Интерфейс начального окна

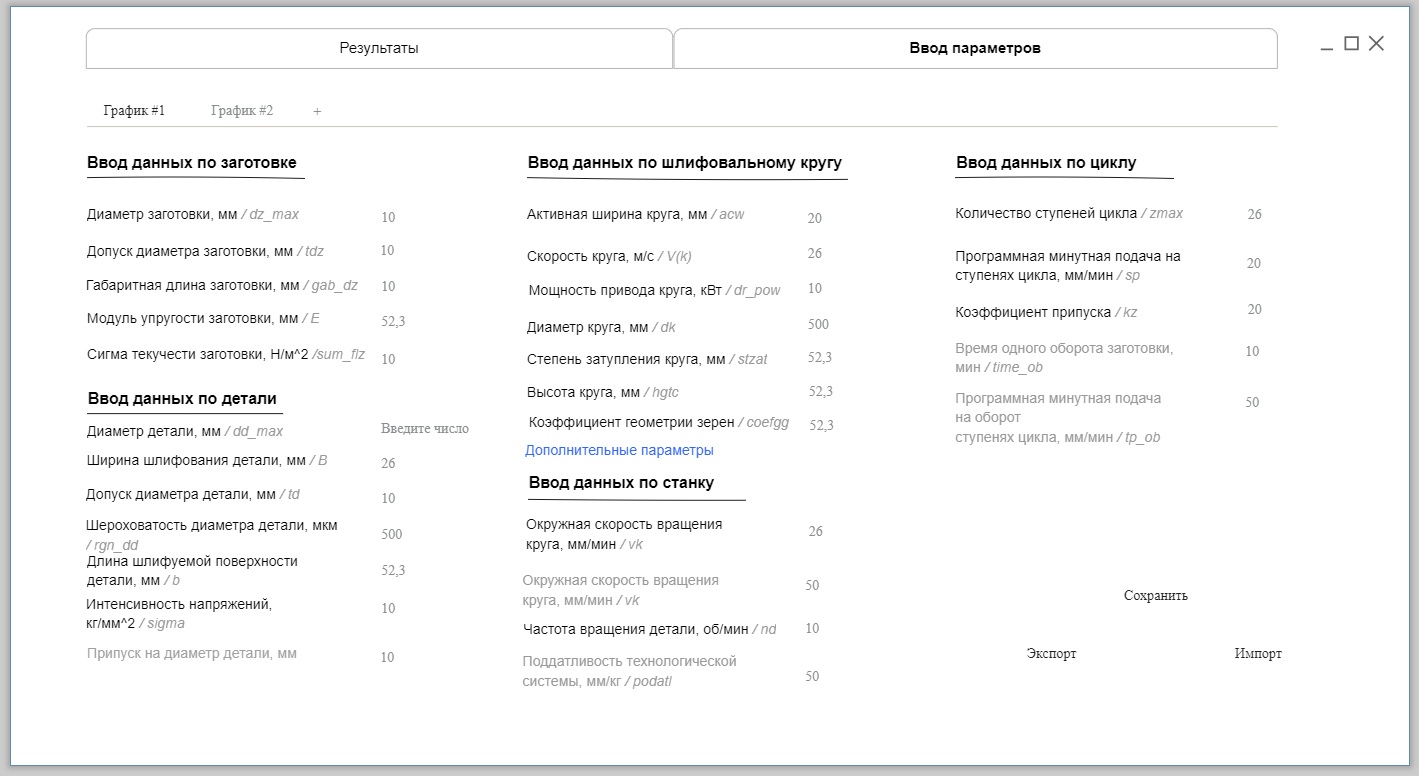


Рисунок 2 - Интерфейс окна ввода параметров

## 2 РАЗРАБОТКА АЛГОРИТМА

Основные сущности в программе:

Класс Data\_Transform для работы с параметрами. В нем 4 метода:

1)calculation для вычисления оптимизированных значений;

2)read\_param для чтения параметров из Excel файла;

3)write\_param для ввода параметров в Excel файл;

4)removing\_param для очистки параметров из Excel файла и самого объекта;

Атрибутом класса является словарь констант с соответствующими названиями.

Схема основного алгоритма программы показана на рисунке 3 и рисунке 4.

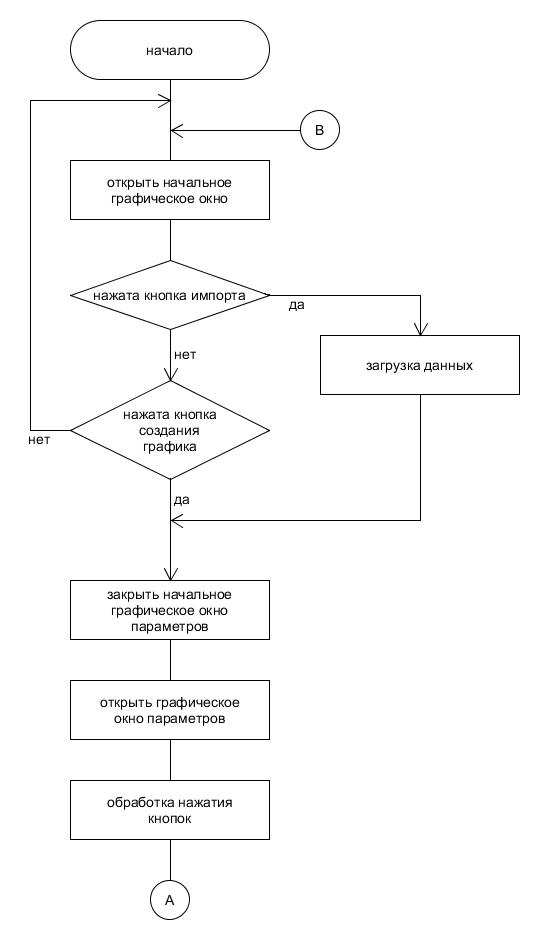


Рисунок 3 – Основной алгоритм программы

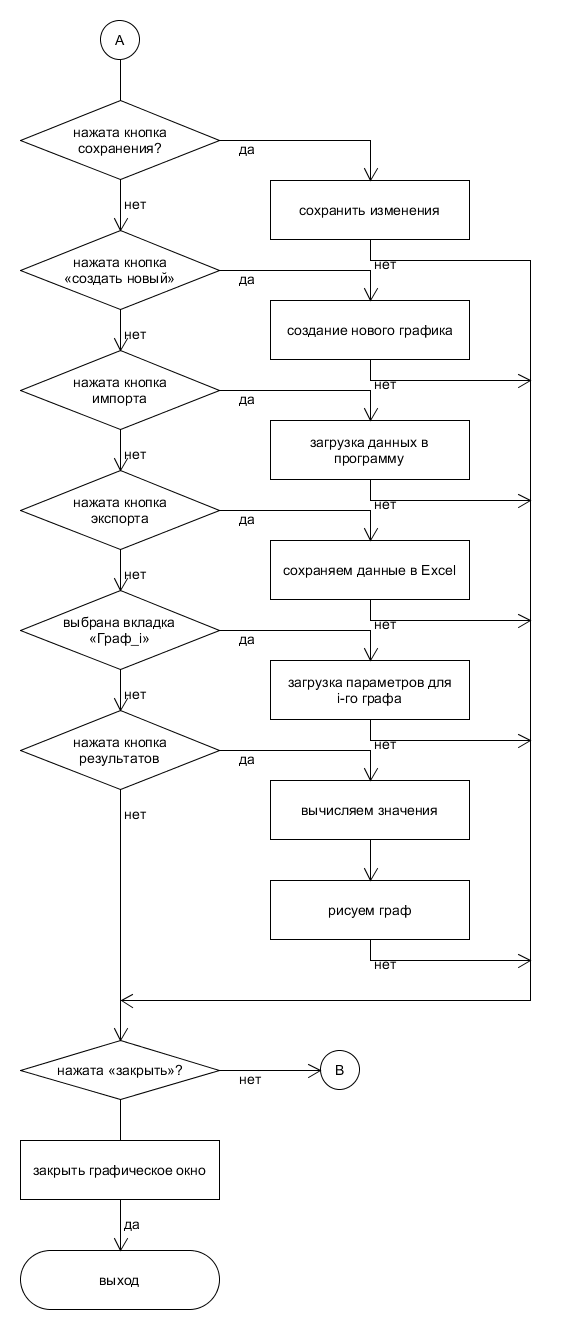


Рисунок 4 – Основной алгоритм программы

## 3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Данная программа предназначена для выполнения оптимизации работы станка ЧПУ. Для работы с ней нужно запустить файл MainWindow.exe. При запуске программы откроется окно (рисунок 5).

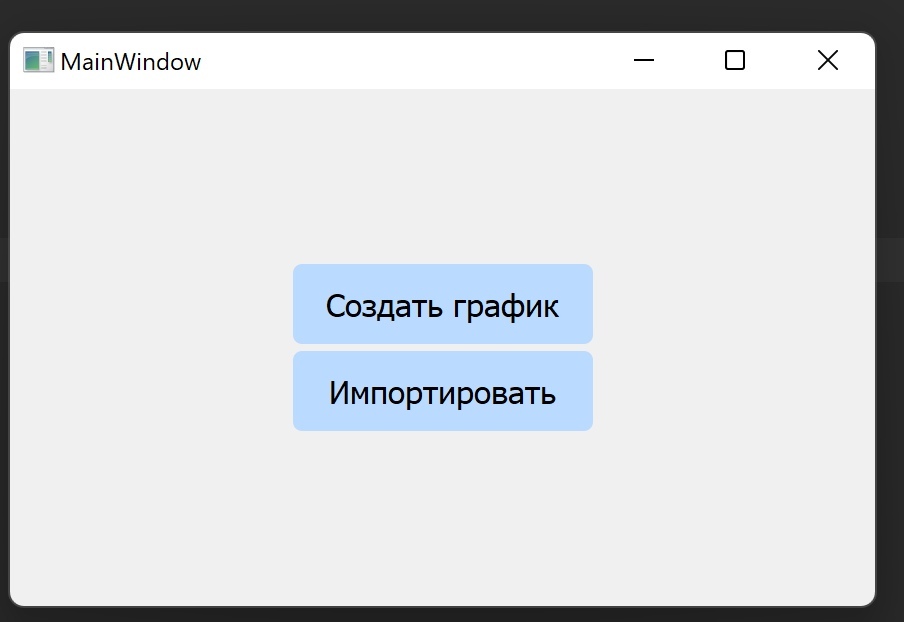


Рисунок 5 – Начальное окно

Пользователь видит 2 кнопки: «Создать график» и «Импортировать». При нажатии второй кнопки пользователь может загрузить начальные данные для станка ЧПУ. При нажатии первой кнопки закроется данное окно и откроется окно (рисунок 6).

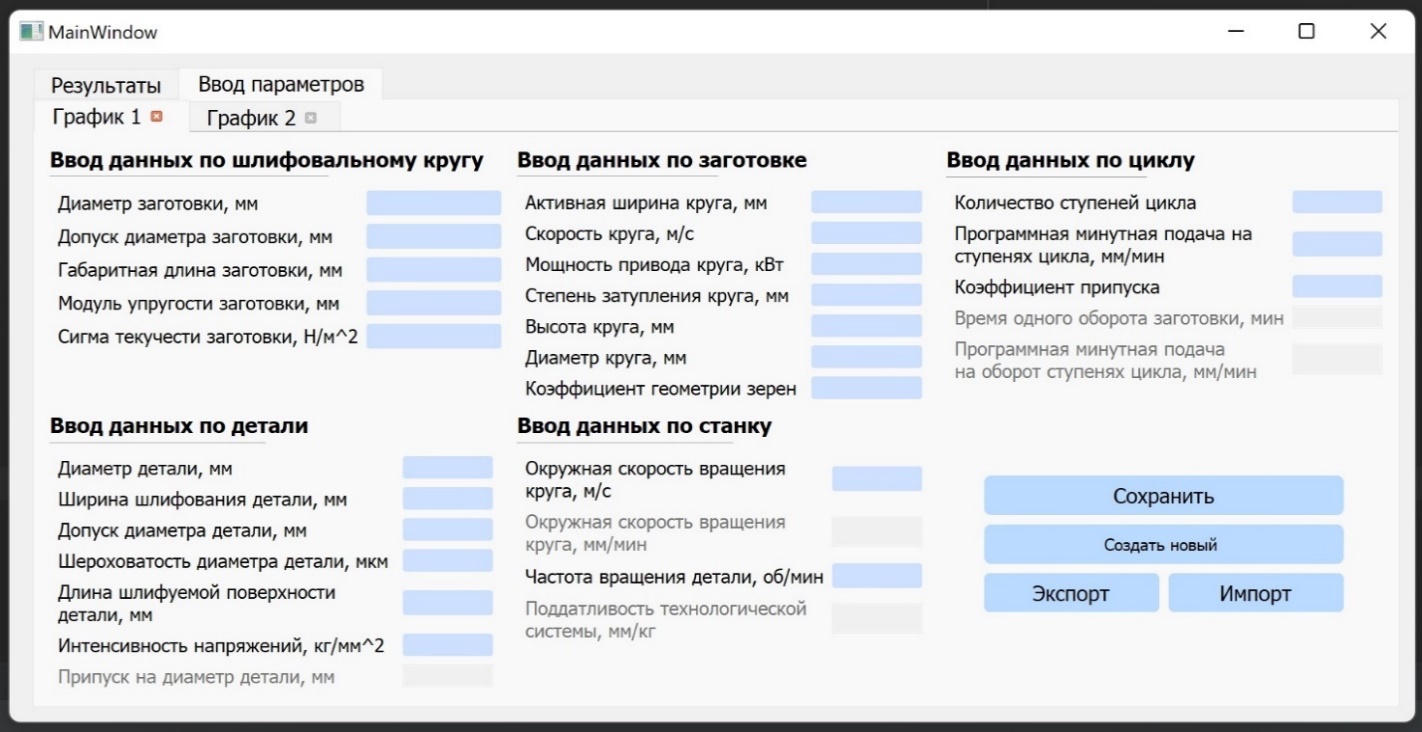


Рисунок 6 – Окно ввода параметров

В центре окна расположены поля для ввода исходных данных станка ЧПУ. В правом нижнем углу есть 4 кнопки. Кнопка «Сохранить» сохраняет введенных пользователем занчения, кнопка «Создать новый» создает новую вкладку «график» с пустыми значения, кнопка «Экспорт» позволяет скачать сохраненные данные в данном графике, кнопка «Импорт» позволяет загрузить начальные данные в данный график.

В левом верхнем углу расположены 4 кнопки. Кнопка «Ввод параметров» открывает вкладку(рисунок 6), кнопки «График 1» и «График 2» открывают открывают поля ввода параментов для построения соответствующего графика. Кнопка «Результаты» показывает построенный график для данный введенных во вкладке «График 1» или «График 2».

Для завершения работы с программой необходимо щелкнуть по кнопке с крестиком в верхнем правом углу.

# ЗАКЛЮЧЕНИЕ

В ходе коллективной работы над проектом были поставлены точные требования к программе, затем были выявлены элементы интерфейса пользователя, разработаны необходимые математические модели, определены и детализированы структуры данных и алгоритмы. После завершения проектирования алгоритмы были реализованы на языке Python. Разработанный код был проверен на контрольных тестах и в код были внесены необходимые исправления. Для программы было разработано руководство пользователя. Таким образом, промежуточные задачи были решены, что позволит в будущем достигнуть цели.

Результаты работы будут использованы в дальнейшем для оптимизации работы станков с ЧПУ.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Лауферман, О. В. Разработка программного продукта: профессиональные стандарты, жизненный цикл, командная работа : учебное пособие / О. В. Лауферман, Н. И. Лыгина. — Новосибирск : НГТУ, 2019. — 75 с. — URL: https://e.lanbook.com/book/152251 (дата обращения: 22.07.2021).

2 Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.

3 Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 990 с

4 Переверзев, П.П. Теория и методика расчета оптимальных циклов обработки деталей на круглошлифовальных станках с программным управлением / П.П. Переверзев – 1999. – с. 1-159.

# ПРИЛОЖЕНИЕ а

А.1 Файл param.py

from PyQt5 import QtCore, QtGui, QtWidgets

import sys

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(865, 517)

MainWindow.setMinimumSize(QtCore.QSize(700, 500))

MainWindow.setIconSize(QtCore.QSize(0, 0))

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

self.gridLayout = QtWidgets.QGridLayout(self.centralwidget)

self.gridLayout.setContentsMargins(0, -1, 0, -1)

self.gridLayout.setVerticalSpacing(7)

self.gridLayout.setObjectName("gridLayout")

self.pushButton\_2 = QtWidgets.QPushButton(self.centralwidget)

sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Maximum)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.pushButton\_2.sizePolicy().hasHeightForWidth())

self.pushButton\_2.setSizePolicy(sizePolicy)

self.pushButton\_2.setMinimumSize(QtCore.QSize(300, 80))

font = QtGui.QFont()

font.setFamily("Tahoma")

font.setPointSize(12)

self.pushButton\_2.setFont(font)

self.pushButton\_2.setStyleSheet("QPushButton\n"

"{\n"

" border-radius: 9px;\n"

" background-color: rgb(187, 218, 255);\n"

"}")

self.pushButton\_2.setObjectName("pushButton\_2")

self.gridLayout.addWidget(self.pushButton\_2, 4, 1, 1, 1)

spacerItem = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem, 0, 1, 1, 1)

self.pushButton = QtWidgets.QPushButton(self.centralwidget)

sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Maximum)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.pushButton.sizePolicy().hasHeightForWidth())

self.pushButton.setSizePolicy(sizePolicy)

self.pushButton.setMinimumSize(QtCore.QSize(300, 80))

font = QtGui.QFont()

font.setFamily("Tahoma")

font.setPointSize(12)

self.pushButton.setFont(font)

self.pushButton.setStyleSheet("QPushButton\n"

"{\n"

" border-radius: 9px;\n"

" background-color: rgb(187, 218, 255);\n"

"}")

self.pushButton.setObjectName("pushButton")

self.gridLayout.addWidget(self.pushButton, 3, 1, 1, 1)

spacerItem1 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem1, 5, 1, 1, 1)

spacerItem2 = QtWidgets.QSpacerItem(68, 55, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)

self.gridLayout.addItem(spacerItem2, 3, 2, 1, 1)

spacerItem3 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)

self.gridLayout.addItem(spacerItem3, 4, 0, 1, 1)

MainWindow.setCentralWidget(self.centralwidget)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "MainWindow"))

self.pushButton\_2.setText(\_translate("MainWindow", "Импортировать"))

self.pushButton.setText(\_translate("MainWindow", "Создать график"))

if \_\_name\_\_ == "\_\_main\_\_":

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec\_())

А.2 ФайлGraf.py

import sys

import openpyxl

import matplotlib

matplotlib.use('Qt5Agg')

from PyQt5 import QtCore, QtGui, QtWidgets

from matplotlib.backends.backend\_qt5agg import FigureCanvasQTAgg, NavigationToolbar2QT as NavigationToolbar

from matplotlib.figure import Figure

class MplCanvas(FigureCanvasQTAgg):

def \_\_init\_\_(self, parent=None, width=5, height=4, dpi=100):

fig = Figure(figsize=(width, height), dpi=dpi)

self.axes = fig.add\_subplot(111)

super(MplCanvas, self).\_\_init\_\_(fig)

class MainWindow(QtWidgets.QMainWindow):

def \_\_init\_\_(self, \*args, \*\*kwargs, ):

super(MainWindow, self).\_\_init\_\_(\*args, \*\*kwargs)

sc = MplCanvas(self, width=5, height=4, dpi=100)

sc.axes.plot(ddo, time\_cicle)

# Create toolbar, passing canvas as first parament, parent (self, the MainWindow) as second.

toolbar = NavigationToolbar(sc, self)

layout = QtWidgets.QVBoxLayout()

layout.addWidget(toolbar)

layout.addWidget(sc)

# Create a placeholder widget to hold our toolbar and canvas.

widget = QtWidgets.QWidget()

widget.setLayout(layout)

self.setCentralWidget(widget)

self.show()

app = QtWidgets.QApplication(sys.argv)

w = MainWindow()

app.exec\_()

А.3 ФайлGraf.py

from PyQt5 import QtWidgets

from main import Ui\_MainWindow

from param import Ui\_ParamWindow

import sys

class mywindow(QtWidgets.QMainWindow):

def \_\_init\_\_(self):

super(mywindow, self).\_\_init\_\_()

self.ui = Ui\_MainWindow()

self.ui.setupUi(self)

app = QtWidgets.QApplication([])

application = mywindow()

application.show()

sys.exit(app.exec())

А.4 Файл api.py

import openpyxl

class Data\_Transform:

def \_\_init\_\_(self, param, row):

self.param = param

self.row = row

def read\_param(self):

f = openpyxl.load\_workbook("datagrafiks2.xlsx")

sheet = f["Первый лист"]

i = 1

for key in self.param:

self.param[key] = sheet.cell(row = self.row, column = i).value()

i += 1

f.save("datagrafiks2.xlsx")

def write\_param(self):

f = openpyxl.load\_workbook("datagrafiks2.xlsx")

sheet = f["Первый лист"]

i = 1

for key in self.param:

sheet.cell(row=self.row, column=i, value=self.param[key])

i += 1

f.save("datagrafiks2.xlsx")

def removing\_param(self):

f = openpyxl.load\_workbook("datagrafiks2.xlsx")

sheet = f["Первый лист"]

for i in range(1, 15):

sheet.cell(row=self.row, column=i, value="")

f.save("datagrafiks2.xlsx")

self.param.clear()