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.

Авторы работы:

Студент группы: ЕТ – 112

Никитин М.С.

Студент группы: ЕТ – 112

Мухутдинов В.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.

Работа защищена с оценкой

Челябинск – 2022

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 3
1 ПОСТАНОВКА ЗАДАЧИ 4
2 РАЗРАБОТКА АЛГОРИТМА 5
3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 8
ЗАКЛЮЧЕНИЕ 10
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 11
ПРИЛОЖЕНИЕ А 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
	По шлифовальному кругу
	По детали
	По заготовке
	По станку
	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

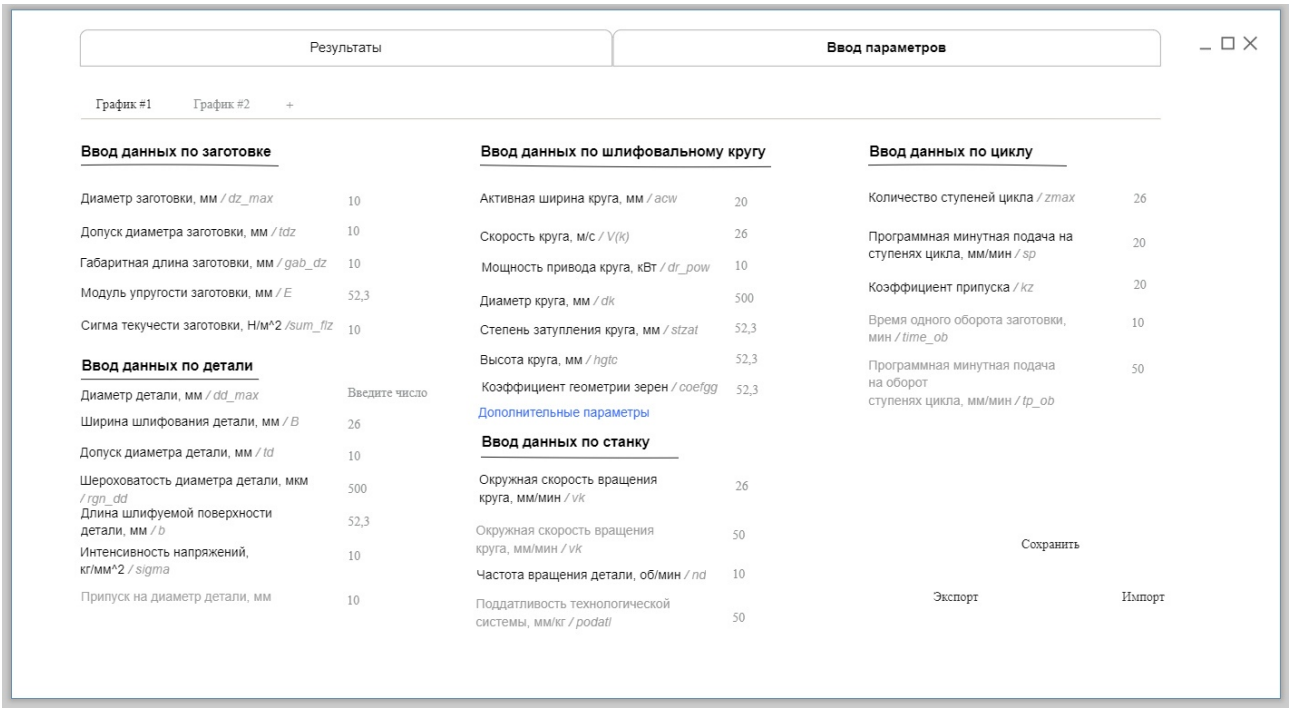

Рисунок 2 - Интерфейс окна ввода параметров

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.

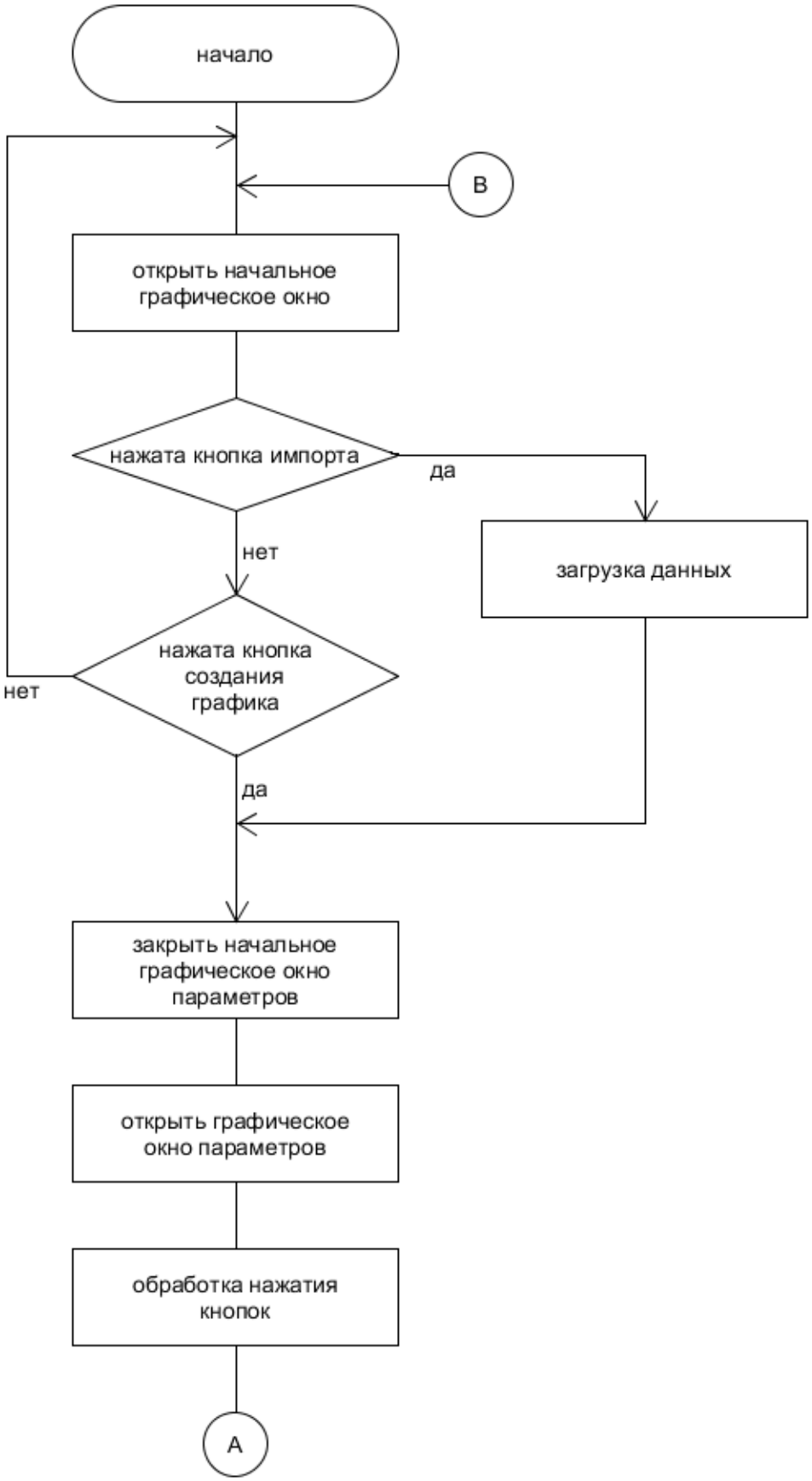

Рисунок 3 – Основной алгоритм программы

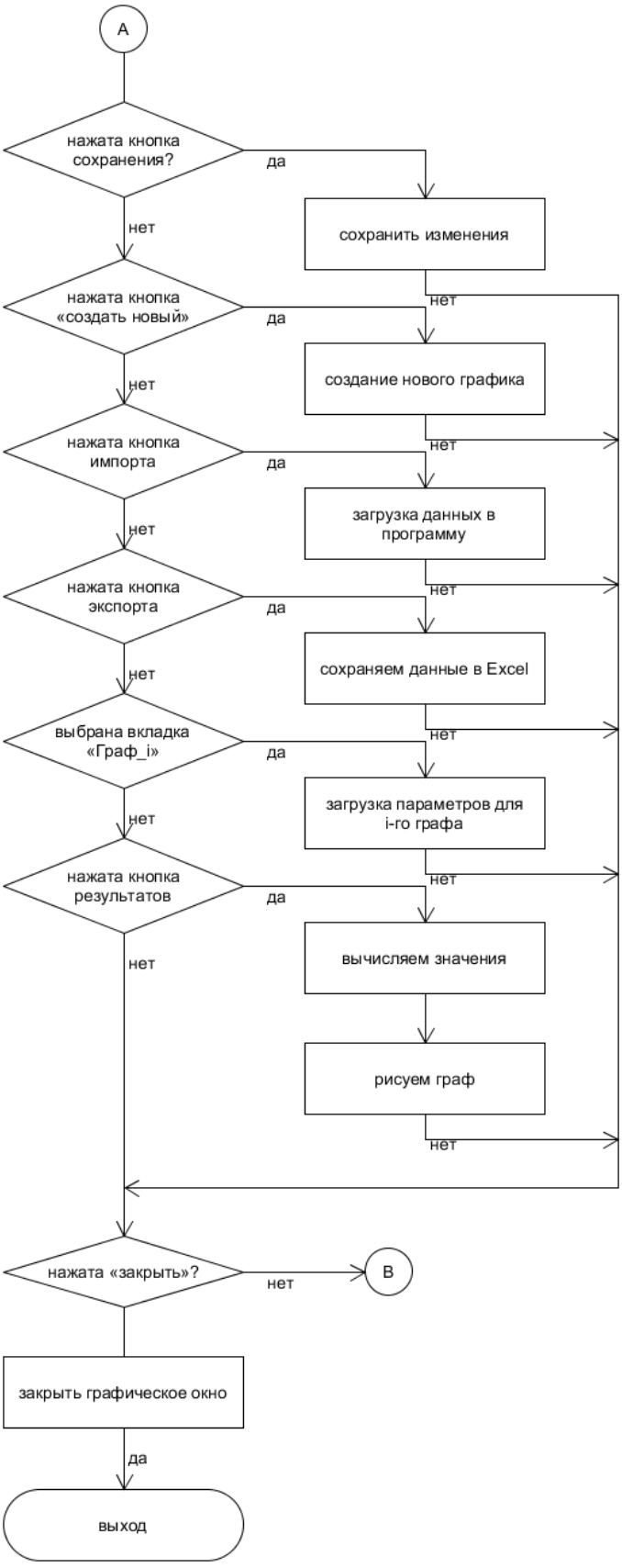

Рисунок 4 – Основной алгоритм программы

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).

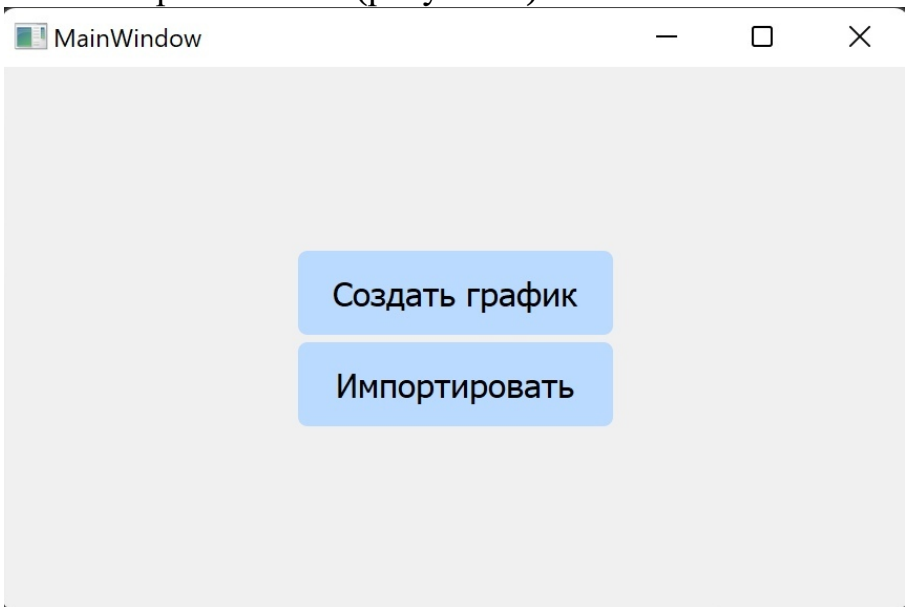

Рисунок 5 – Начальное окно

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).

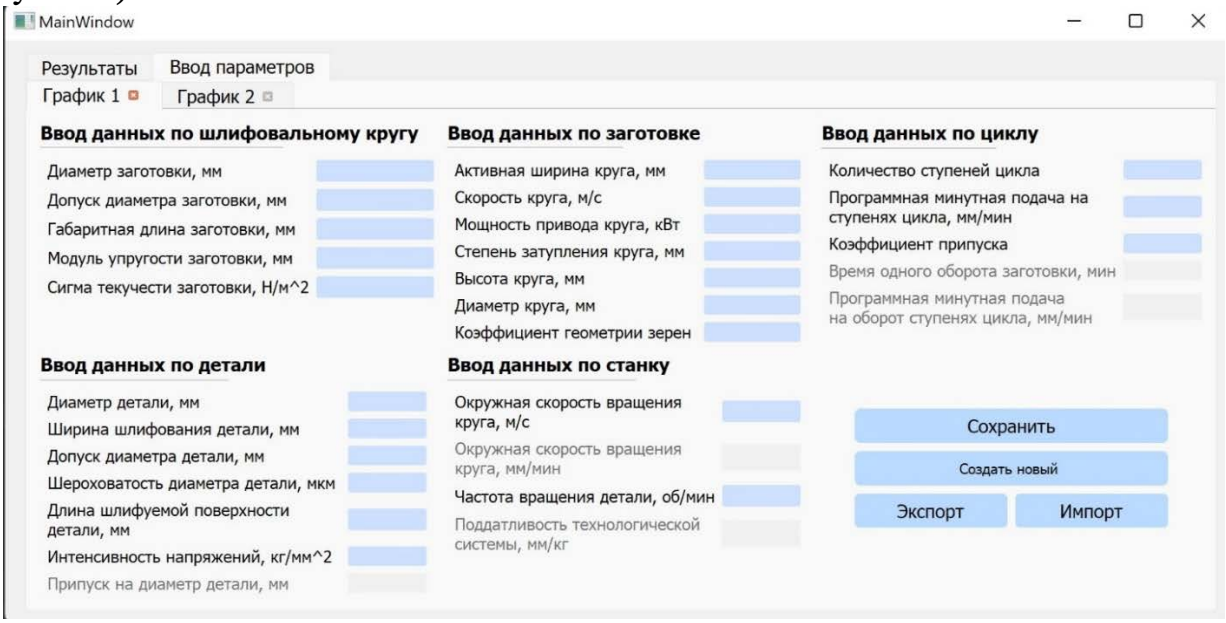

Рисунок 6 – Окно ввода параметров

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 парамментов 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

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).

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

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

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

A.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_())
```

A.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_()
```

A.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())
```

A.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")

i
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")
s
h
e
for key in self.param:
t           sheet.cell(row=self.row, column=i, value=self.param[key])
            i += 1
f["Первый лист"]
        f.save("datagrafiks2.xlsx")
    def removing_param(self):
        f = openpyxl.load_workbook("datagrafiks2.xlsx")
s       sheet = f["Первый лист"]
h       for i in range(1, 15):
e           sheet.cell(row=self.row, column=i, value="")
e       f.save("datagrafiks2.xlsx")
t       self.param.clear()
f["Первый лист"]
```