Российская Федерация

ПРОГРАММА ДЛЯ ЭВМ

ПРОГРАММА ДЛЯ ПРЕОБРАЗОВАНИЯ АННОТАЦИЙ К ИЗОБРАЖЕНИЯМ

Фрагменты исходного текста программы

Листов 3

Авторы:

Юденич Александр Анатольевич

Колузов Андрей Владимирович

Патрушев Сергей Михайлович

Российская Федерация

Анапа

2020 г

РАСПЕЧАТКА ИСХОДНОГО ТЕКСТА ПРОГРАММЫ

Файл AnnotationConverterWidget.py

```
from PyQt5 import QtWidgets, QtGui, QtCore
import json
import os
import cv2 as cv


class AnnotationConverterWidget(QtWidgets.QWidget):

    @staticmethod
    def convert_annotations(annotations, images_folder):
        result = {}
        for img in annotations.values():
            image_file_name = img["filename"]
            image_file_path = os.path.join(images_folder, image_file_name)
            image = cv.imread(image_file_path)
            image_height, image_width, _ = image.shape
            shape = img["regions"][0]["shape_attributes"]
            polygon_points = list(zip(shape["all_points_x"], shape["all_points_y"]))
            image_data = {
                "image_size": {
                    "width": image_width,
                    "height": image_height
                },
                "polygon_points": polygon_points
            }
            result[image_file_name] = image_data
        return result

    def btn_load_annotations_clicked(self):
        annotations_file_name = QtWidgets.QFileDialog.getOpenFileName(
            self, "Выберите файл с аннотациями",
            QtCore.QDir.currentPath(),
            "json (*.json)"
        )[0]

        if annotations_file_name:
            self.lbl_annotations_file.setText("Файл с аннотациями: " + annotations_file_name)
            self.annotation_file = annotations_file_name
            with open(annotations_file_name) as annotations_json:
                data = json.load(annotations_json)
                formatted_json = json.dumps(data, indent=4)
                self.text_input_annotations.setText(formatted_json)

    def btn_choose_images_folder_clicked(self):
        images_directory = QtWidgets.QFileDialog.getExistingDirectory(
            self, 'Выберите папку с изображениями',
            QtCore.QDir.currentPath())

        if images_directory:
            self.lbl_images_folder.setText("Папка с изображениями: " + images_directory)
            self.images_folder = images_directory

    def btn_convert_annotations_clicked(self):
        if self.images_folder is None:
            QtWidgets.QMessageBox.information(
                self,
                'Информация',
                'Папка с изображениями не выбрана')
```

```
            return

        try:
            data = json.loads(self.text_input_annotations.toPlainText())
            result = self.convert_annotations(data, self.images_folder)
            formatted_json = json.dumps(result, indent=4)
            self.text_output_annotations.setText(formatted_json)
        except:
            QtWidgets.QMessageBox.information(
                self,
                'Информация',
                "Не удалось выполнить конвертацию.\n"
                "Проверьте корректность аннотаций.")

    def btn_save_to_file_clicked(self):
        annotations_file_name = QtWidgets.QFileDialog.getSaveFileName(
            self, "Выберите файл для сохранения",
            QtCore.QDir.currentPath(),
            "json (*.json)"
        )[0]
        if annotations_file_name:
            annotations = json.loads(self.text_output_annotations.toPlainText())
            with open(annotations_file_name, "w") as file:
                json.dump(annotations, file)

    def __init__(self, parent=None):
        super().__init__(parent)
        self.setWindowTitle('Технополис "ЭРА"')
        self.setWindowIcon(QtGui.QIcon('images/logo_era.ico'))
        self.setWindowState(QtCore.Qt.WindowMaximized)

        self.annotation_file = None
        self.images_folder = None

        btn_load_annotations = QtWidgets.QPushButton("Аннотации")
        btn_choose_images_folder = QtWidgets.QPushButton("Изображения")
        btn_convert_annotations = QtWidgets.QPushButton("Конвертировать")
        btn_save_to_file = QtWidgets.QPushButton("Сохранить в файл")

        btn_load_annotations.clicked.connect(self.btn_load_annotations_clicked)
        btn_convert_annotations.clicked.connect(self.btn_convert_annotations_clicked)
        btn_choose_images_folder.clicked.connect(self.btn_choose_images_folder_clicked)
        btn_save_to_file.clicked.connect(self.btn_save_to_file_clicked)

        self.lbl_annotations_file = QtWidgets.QLabel("Файл с аннотациями: не выбран")
        self.lbl_images_folder = QtWidgets.QLabel("Папка с изображениями: не выбрана")

        self.text_input_annotations = QtWidgets.QTextEdit()
        self.text_output_annotations = QtWidgets.QTextEdit()

        layout_buttons = QtWidgets.QHBoxLayout()
        layout_buttons.setAlignment(QtCore.Qt.AlignLeft)
        layout_buttons.addWidget(btn_load_annotations)
        layout_buttons.addWidget(btn_choose_images_folder)
        layout_buttons.addWidget(btn_convert_annotations)
        layout_buttons.addWidget(btn_save_to_file)

        layout_labels = QtWidgets.QVBoxLayout()
        layout_labels.setAlignment(QtCore.Qt.AlignLeft)
        layout_labels.addWidget(self.lbl_annotations_file)
        layout_labels.addWidget(self.lbl_images_folder)

        layout_text_editors = QtWidgets.QHBoxLayout()
```

```
        layout_text_editors.addWidget(self.text_input_annotations)
        layout_text_editors.addWidget(self.text_output_annotations)

        layout_main = QtWidgets.QVBoxLayout()
        layout_main.addLayout(layout_buttons)
        layout_main.addLayout(layout_labels)
        layout_main.addLayout(layout_text_editors)

        self.setLayout(layout_main)
```

Файл main.py

```
import sys
from PyQt5 import QtWidgets
from AnnotationConverterWidget import AnnotationConverterWidget

if __name__ == '__main__':
    app = QtWidgets.QApplication(sys.argv)
    app.setStyle(QtWidgets.QStyleFactory.create("Fusion"))
    aerial_roads_widget = AnnotationConverterWidget()
    aerial_roads_widget.show()
    sys.exit(app.exec())
```