МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Пензенский государственный технологический университет»**

(ПензГТУ)

Факультет автоматизированных информационных технологий

Кафедра «Программирование»

Дисциплина «Программирование»

**Отчет по лабораторной работе №2**

**«Работа с PyQT»**

Выполнил:

Студент гр. 24ИП3б Цыбин Д.Н.

Проверил:

ассистент кафедры «Программирование»

Зупарова В.В.

**Лабораторное задание (№1 простой сложности)**

Напишите программу «Заказ в Макдональдсе» с графическим пользовательским интерфейсом на PyQT.

Пользователь должен иметь возможность выбирать одно или несколько блюд. После нажатия на кнопку «Заказать» в отдельном виджете должен отображаться «чек» с выбранными позициями.

Класс, реализующий окно приложения назовите MacOrder. Для выбора позиций используйте виджет QCheckBox. Храните список из чекбоксов в атрибуте menu\_checkboxes объекта класса приложения.

Позиции в меню: "Чизбургер", "Гамбургер", "Кока-кола", "Наггетсы".

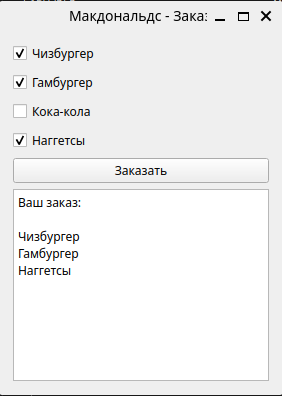
Кнопка заказа должна называться order\_btn.

Заказ должен выводится в виджет QPlainTextEdit с названием result. Позиции в заказе должны выводится в том же порядке, как в меню. Текст заказа должен начинаться с «Ваш заказ:» и пустой строки.

**Листинг**



**Примеры входных-выходных данных**



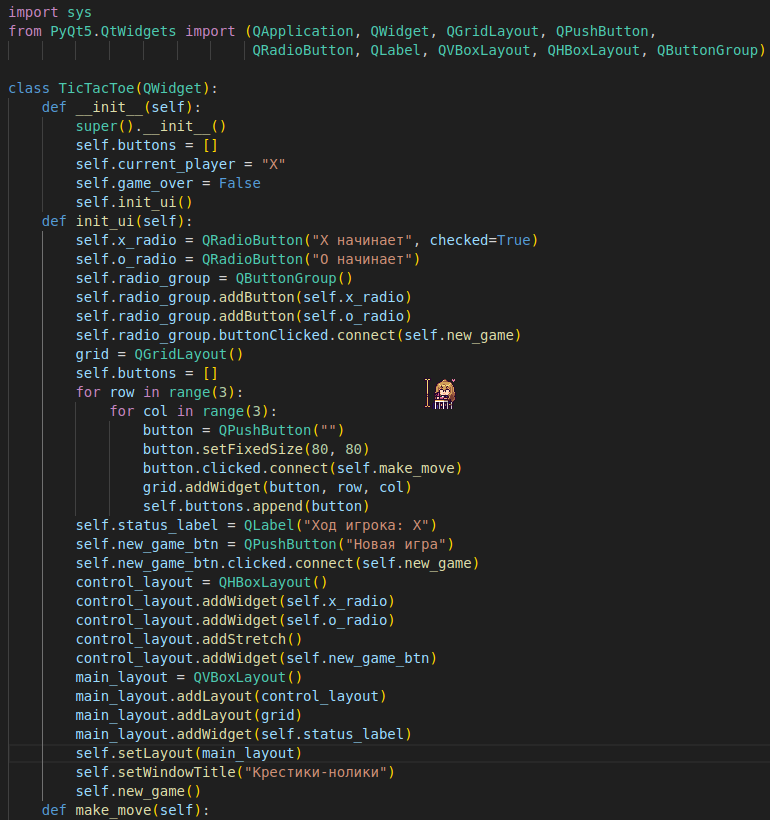
**Лабораторное задание (№2 простой сложности)**

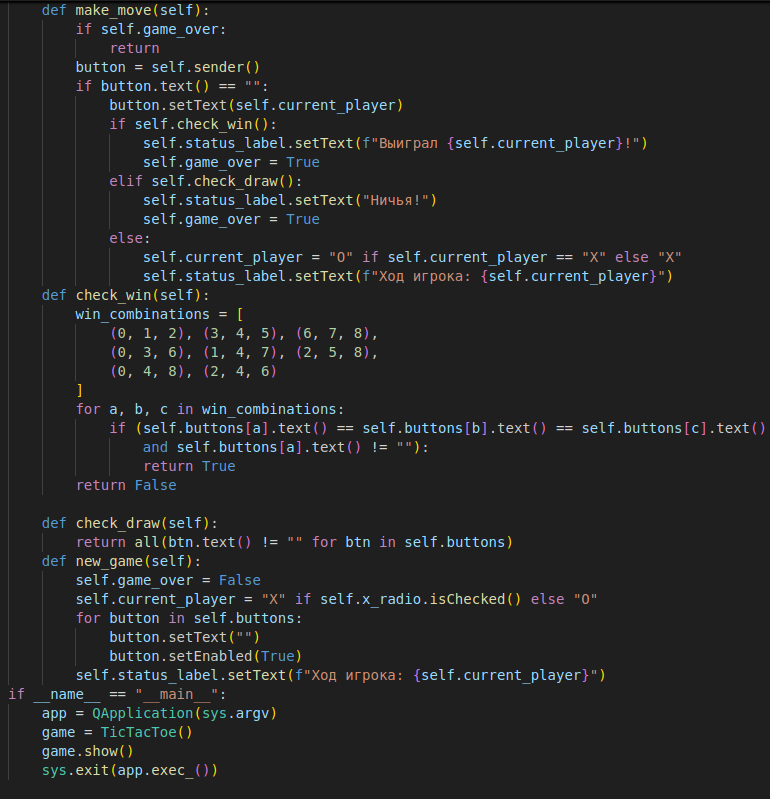
Разработайте игру «Крестики-нолики» на поле 3×3 с графическим пользовательским интерфейсом на PyQT.

Поле должно быть формой (QWidget) с набором кнопок (QPushButton) ( подсказка: виджеты можно объединять в списки).

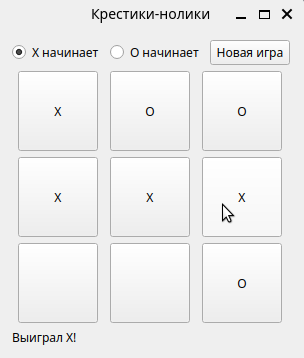
При нажатии на кнопку в зависимости от того, чей ход, текст кнопки меняется на 'X' или 'O'. Выбор первого игрока производится с помощью QRadioButton (по умолчанию первых игрок X). В случае победы или ничьей должно отображаться соответствующее сообщение. Для отображения надо использовать виджет QLabel. В случае победы X отображать "Выиграл X!", в случае победы O отображать " Выиграл O!", в случае ничьей отображать "Ничья!". После победы одного из игроков игровое поле нужно блокировать до начала новой игры. Новая игра может быть начата после нажатия на соответствующую кнопку (QPushButton) или при смене обозначения хода у первого игрока (QRadioButton).

**Листинг**





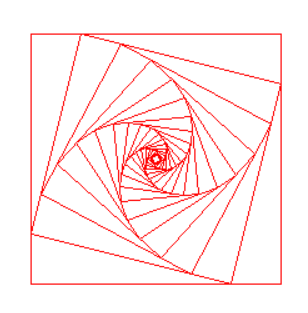
**Примеры входных-выходных данных**



**Лабораторное задание (№1 средней сложности)**

Напишите программу «Квадрат-объектив — 2». На экране нарисован квадрат. Пользователь в соответствующие поля вводит вещественное число k (< 1) — коэффициент масштабирования стороны и целое число n — количество повторений.

Каждая сторона квадрата делится в пропорции, соответствующей указанному коэффициенту, полученные точки соединяются и получается новый квадрат. Данная операция повторяется n раз. Например, если задать k = 0.8, а n = 20, результат должен получиться такой:



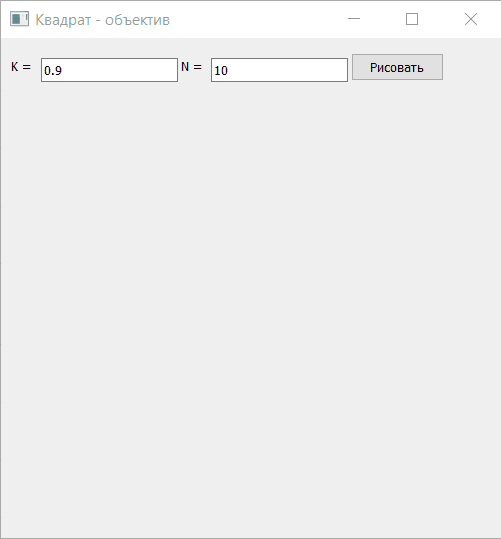
Самый большой квадрат должен идти из точки с координатами (150, 150) и иметь сторону 200

Класс, реализующий окно приложения, назовите Square2.

Поле для ввода числа k назовите k. Для числа n соответственно n

Цвет, которым вы будете рисовать квадраты сохраните в атрибуте color объекта класса приложения при инициализации в виде QColor.

Кнопку назовите draw

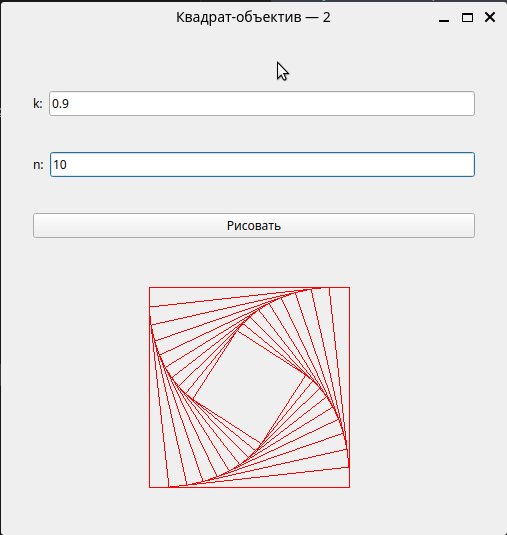


Размер окна приложения сделайте строго 500 пикселей по каждому измерению. Для изображения квадрата воспользуйтесь методом [drawPolygon](https://doc.qt.io/qtforpython-5/PySide2/QtGui/QPainter.html#PySide2.QtGui.PySide2.QtGui.QPainter.drawPolygon) класса [QPainter](https://doc.qt.io/qtforpython-5/PySide2/QtGui/QPainter.html). Для создания объекта многоугольника, который передается методу drawPolygon, используйте класс [QPolygonF](https://doc.qt.io/qtforpython-5/PySide2/QtGui/QPolygonF.html#more). Для добавления точек в многоугольник используйте метод [append](https://doc.qt.io/qtforpython-5/PySide2/QtGui/QPolygonF.html#PySide2.QtGui.PySide2.QtGui.QPolygonF.append). Все вычисления производите в вещественных числах, вещественные результаты вычислений передавайте конструктору класса [QPointF](https://doc.qt.io/qtforpython-5/PySide2/QtCore/QPointF.html#PySide2.QtCore.PySide2.QtCore.QPointF) - это точки - вершины многоугольника.

**Листинг**



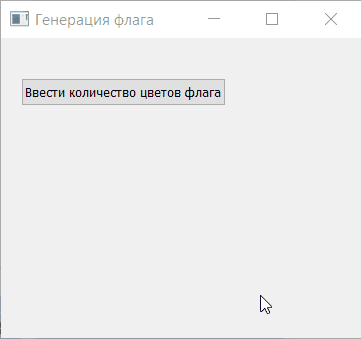
**Примеры входных-выходных данных**



**Лабораторное задание (№2 средней сложности)**

Напишите программу с графическим пользовательским интерфейсом на PyQt, которая создаёт изображение разноцветного (полосатого) флага. Количество цветов необходимо получить от пользователя с помощью диалогового окна.

Цвета генерируются случайно.



Минимальное число, которое разрешено вводить пользователю должно быть 1, а максимальное 10. По умолчанию поставьте число 3 и сделайте, чтобы шаг был равен 1

Генерировать цвета обязательно с помощью **random.randrange**. Для получения одного случайного цвета обязательно сгенерировать три числа, отвечающие за r, g и b каналы (для каждого канала от 0 до 256, не включая 256). Цвета нужно генерировать для прямоугольников сверху вниз.

Класс, реализующий окно приложения, назовите **RandomFlag**.

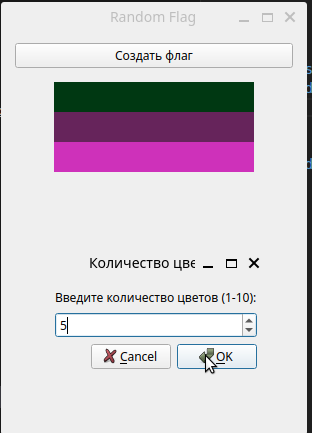
При инициализации приложения необходимо сохранить параметры прямоугольника (одной полосы флага) в атрибут **base** объекта класса приложения в виде списка, первые 2 значения в котором - это x и y координата левой верхней точки верхней полосы флага, а следующие два - величины сторон (длина и ширина)

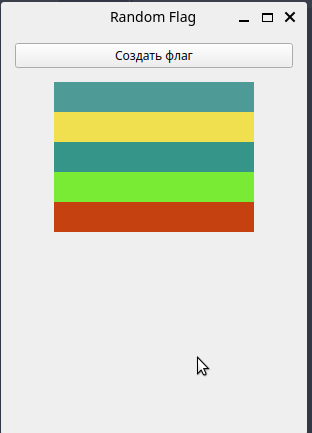
Кнопку назовите **button**. Размер окна приложения сделайте строго 300 на 400 пикселей.

**Листинг**



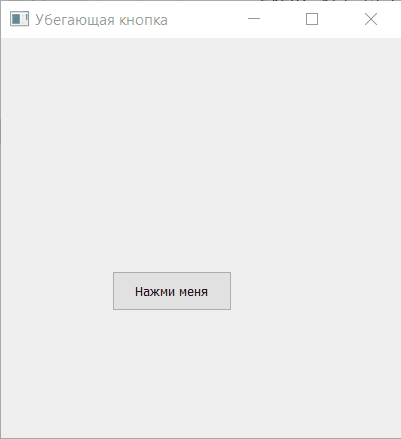
**Примеры входных-выходных данных**





**Лабораторное задание (№1 высокой сложности)**

Напишите программу «Убегающая кнопка» с использованием виджетов PyQT. На форме произвольного размера располагается кнопка. Когда курсор мыши приближается к кнопке, она перемещается таким образом, чтобы нельзя было на неё нажать. Кнопка не должна исчезать с экрана (частично или полностью).



Назовите главный класс программы MyWidget.

У кнопки должно быть название button.

**Листинг**



**Примеры входных-выходных данных**

К отчёту по лабораторной работе будет приложено видео.

