

**ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП****ЭКЗАМЕНАЦИОННОЕ ЗАДАНИЕ****ПО НАПРАВЛЕНИЮ****«ПРОГРАММИРОВАНИЕ»****ВАРИАНТ 2.****ЗАДАНИЕ:**

Вам предстоит написать программу в виде графического редактора. В данном задании представлены общие требования к программе, интерфейсу и основной логика.

**ОБЩИЕ ТРЕБОВАНИЯ:**

Написать программу согласно представленному заданию. В программе должны быть реализованы указанные в задании особенности программы, макет интерфейса и требование к коду.

При запуске программы открывается окно. В окне размещена палитра фигур и графическое поле. Выбрав фигуру, пользователь может с помощью левой кнопки мыши нарисовать её на графическом поле, задав в процессе размер. Размещённые на поле фигуры можно выбрать, нажав по ним левой кнопкой мыши. Пользователь может изменять положение выбранной фигуры на поле.

В ходе оценки проверяется работоспособность модулей программы и использование объектно-ориентированного программирования (для этого следует продемонстрировать инкапсуляцию, наследование и полиморфизм), соблюдение стиля кода, его читаемость и наличие комментариев.

**ОПИСАНИЕ ИНТЕРФЕЙСА**

- Палитра фигур — расположена в левой части окна, содержит изображение трёх фигур, нажав на которые, пользователь выбирает какую фигуру рисовать следующей.
- Графическое поле — занимает всю оставшуюся область экрана, используется для размещения и взаимодействия с фигурами.
- Кнопка Delete — используется для удаления выбранных фигур с графического поля.
- Кнопка Exit — завершает работу программы. Выделена ярким цветом.



**МАКЕТ:**

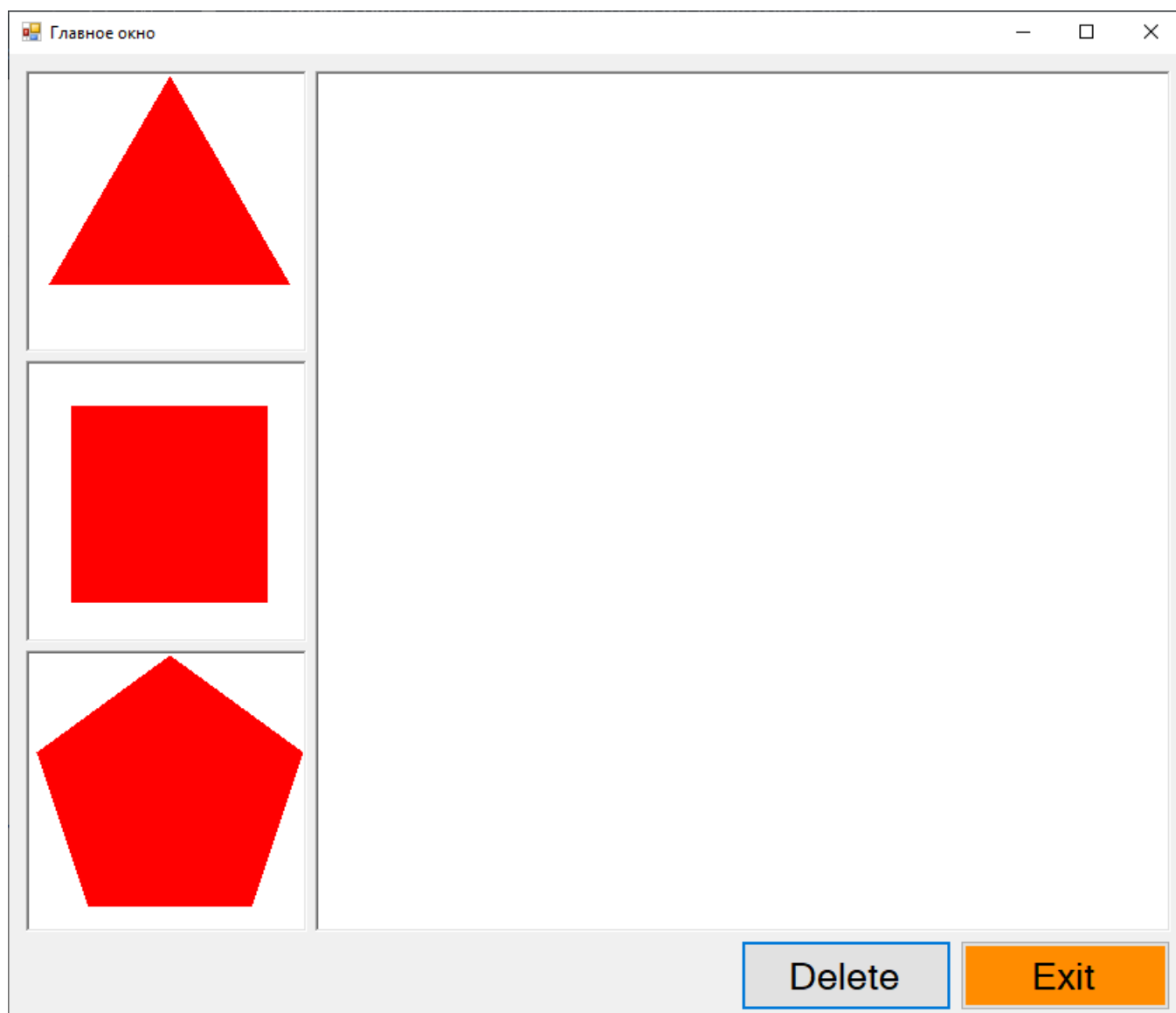


Рис. 1.



## **ЛОГИКА РАБОТЫ ПРОГРАММЫ:**

Пользователь имеет возможность выбрать один из многоугольников, представленных в палитре:

треугольник;

шестиугольник;

семиугольник.

Выбрав нужный многоугольник, пользователь может разместить его на графическом поле.

Единоновременно на графическом поле может быть не больше 5 многоугольников.

Создание многоугольника возможно как с размером по умолчанию, так и с тем размером, который пользователь задаёт самостоятельно. В случае, если был выбран не тот многоугольник, есть возможность отменить выбор.

При отсутствии выбранного многоугольника нажатие на графическое поле позволяет выбрать один из уже нарисованных многоугольников. В этом случае будет выбран многоугольник, чьи координаты центра находятся ближе всего к координатам точки нажатия курсором на графическое поле и расстояние до которых меньше, чем радиус описанной окружности многоугольника (расстояние от координат центра до вершины многоугольника).

Выбранный многоугольник выделяется с помощью отрисовки квадратов по координатам вершин многоугольника. При нажатии кнопки или клавиши Delete выбранный многоугольник удаляется.

Нажатие на кнопку Exit завершает работу программы.

Имеется возможность редактирования выбранного многоугольника. Пользователь может изменить позицию центральной координаты многоугольника, «перетаскив» его на новое место.

**УПРАВЛЕНИЕ:**

Палитра:

- При нажатии левой кнопкой мыши по палитре происходит выбор многоугольника для рисования.

Графическое поле:

- Нажатие на правую кнопку мыши отменяет выбор многоугольника из палитры.
- В случае, если многоугольник из палитры выбран, нажатие по графическому полю левой кнопкой мыши создаёт на нём данный многоугольник.
- Если при создании многоугольника зажать левую кнопку мыши, то появляется возможность задать размер многоугольника (при этом будущий многоугольник постоянно отображается и динамически меняет размер).
- В случае отсутствия выбранного многоугольника нажатие левой кнопкой мыши в графическом поле позволяет выбрать один из многоугольников, уже размещённых в этом поле. Выбран будет многоугольник, чьи координаты центра ближе всего к координатам курсора в момент нажатия левой кнопкой мыши на графическое поле, при этом расстояние должно быть не больше расстояния от центра до вершин многоугольника (радиус описанной окружности).
- При наличии выбранного многоугольника, нажатие клавиши Delete на клавиатуре (или кнопки Delete пользовательского интерфейса) позволяет удалить данный многоугольник из графического поля.
- Нажатие на кнопку Exit завершает работу программы.
- При наличии выбранного многоугольника, зажатие левой кнопки мыши позволяет производить манипуляции с многоугольником на графическом поле, в данном случае перемещение.
- Для отмены выбора многоугольника, пользователь может совершить нажатие левой кнопки мыши вне радиуса описанной окружности всех размещённых на графическом поле многоугольников.



## **ОПИСАНИЕ КЛАССОВ**

При проектировании вы можете сами решать, как реализовывать классы, их методы и поля данных. Ниже приведен рекомендуемый список.

Просим обратить внимание, что данный список содержит минимальный требуемый набор членов класса.

Графическое поле

- Поля данных:
  - Список всех многоугольников.
- Методы:
  - Отрисовка поля;
  - Обработка нажатия левой кнопки мыши.

Многоугольник

- Поля данных:
  - Координаты центра многоугольника.
- Методы:
  - Отрисовка многоугольника;
  - Изменение размера;
  - Перемещение;
  - Отрисовка выделения.

5

## **ОТРИСОВКА МНОГОУГОЛЬНИКОВ:**

В рамках задания требуется написать программу, в функционал которой входит отрисовка правильных многоугольников. Далее представлен возможный вариант решения данной задачи:

У многоугольника имеется заданный размер и координаты центра. Координаты центра многоугольника совпадают с координатами центра описанной окружности многоугольника. Размер многоугольника совпадает с радиусом описанной окружности многоугольника.

Тогда для отрисовки правильного многоугольника требуются точки на окружности с шагом  $2\pi / n$ , где  $n$  – количество вершин многоугольника. Координаты вершин можно рассчитать по формуле:

$$x = r * \cos(\alpha * i) + \text{center\_x}$$

$$y = r * \sin(\alpha * i) + \text{center\_y}$$

Где  $x$ ,  $y$  – координаты вершины многоугольника,  $\text{center\_x}$ ,  $\text{center\_y}$  – координаты центра многоугольника,  $\alpha$  – шаг угла между вершинами,  $i$  – номер вершины.

Также стоит отметить, что в большинстве сред разработки точка отсчёта координат отмечается в левом верхнем углу окна приложения, при этом ось координат  $y$  направлена вниз.



## ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ИНТЕРФЕЙСА

- Размеры элементов интерфейса:
  - размеры кнопок не менее 150 пикселей в ширину и 50 пикселей в высоту;
  - размеры элементов в палитре не менее 200 пикселей в ширину и высоту;
  - размеры графической области не менее 600 пикселей в ширину и высоту.
- Шрифт:
  - размер шрифта в элементах управления не менее 18 пунктов;
  - шрифт имеет черный цвет.
- Цвет:
  - кнопка выхода из программы выделена ярким цветом;
  - фон элементов палитры и графической области белый;
  - многоугольники имеют яркий, легко различимый цвет.

## ТРЕБОВАНИЯ К ПРОЕКТУ:

Участник вправе выбрать язык программирования и среду разработки из предлагаемого списка:

**Языки программирования:** Java, C++, C#, Python, Object Pascal;

**Среды разработки:** IntelliJ IDEA, Processing3, Visual Studio, PyCharm, Pascal ABC.Net, PythonIDE, Lazarus.

6

В программе должны быть выполнены все указанные в задании механики и элементы. Должны быть предусмотрены проверки для предотвращения ошибок работы программы (пример: попытка использовать неинициализированный объект). В случае возникновения ошибок должна быть реализована обратная связь (сообщение с указанием произошедшей ошибки).

Имена переменных и элементов должны иметь говорящие названия на английском языке (соблюдая стиль кода, пример: camelCase, snake\_case). Для написанных подпрограмм и переменных следует оставлять поясняющие комментарии.

Шаблоны комментариев:

Все комментарии – строчные и оформлены по следующему макету:

- Данный метод принимает следующие значения: *имя параметра – его описание* [*; следующий параметр*].
- Выполняет следующие действия: описание функционала.
- В качестве результата возвращает следующее значение: описание результата.

Программа должна быть оптимизирована и рассчитана на длительные сеансы использования.

Прим.: повторное использование созданных объектов, либо их удаление при отсутствии потребности в их использовании.