



ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП

ЭКЗАМЕНАЦИОННОЕ ЗАДАНИЕ

по компетенции

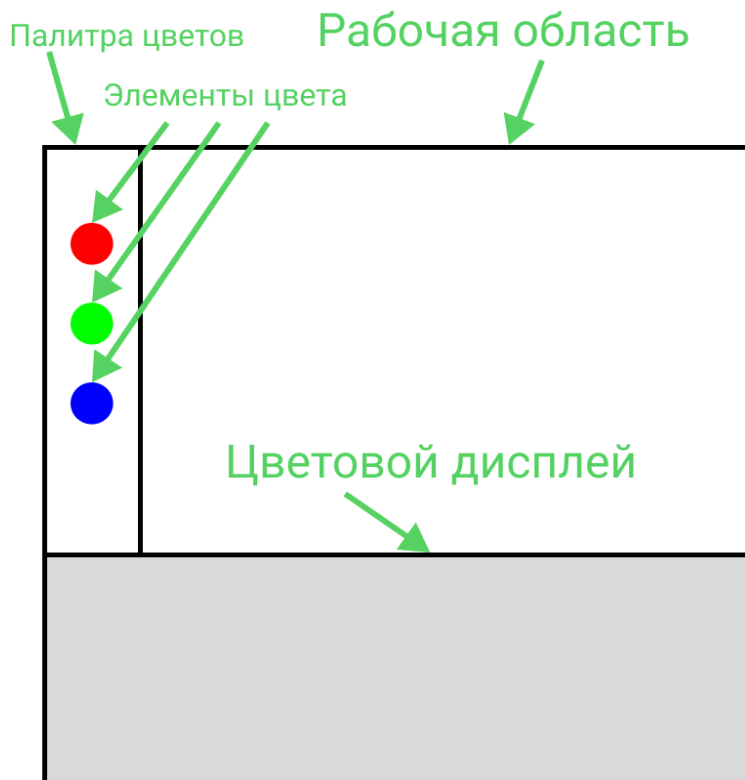
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ. РАЗРАБОТКА ПРИЛОЖЕНИЙ»

Вариант 1.

Создать программу с интерфейсом для составления палитры, согласно заданию. В рамках задания, в программе существуют не менее четырёх сущностей:

- палитра готовых цветов;
- элемент цвета;
- рабочая область;
- цветовой дисплей.

Макет интерфейса:





Палитра — изначально содержит 3 элемента цвета: красный(255, 0, 0), синий(0, 0, 255), зеленый(0, 255, 0). При нажатии левой кнопкой мыши по элементу цвета, создаётся экземпляр данного элемента на рабочей области. При нажатии клавиши «D» или «Пробел» происходит удаление всех добавленных в палитру элементов цвета, кроме исходных.

Элемент цвета — геометрический примитив фиксированного цвета, заданного значениями RGB. Взаимодействие с элементами, которые находятся на рабочем поле определяет окрас цветового дисплея.

Рабочая область — место размещения элементов цвета для взаимодействия.

Цветовой дисплей — область окна прямоугольной формы, чей цвет меняется в зависимости от элементов цвета на рабочем поле. При нажатии левой кнопки мыши размещает в палитре элемент цвета с текущим окрасом цветового дисплея.

Взаимодействие с интерфейсом:

- Обработка нажатия кнопок мыши.
- Перетаскивание объектов.
- Нажатие клавиш.

Проверяемые навыки ООП:

- Применение инкапсуляции для ограничения доступа.
- Использование полиморфизма (преобразование типов).
- Использование наследования.



ФУНКЦИОНАЛ

В рамках задания, для каждого элемента, вам необходимо реализовать следующий функционал:

Палитра готовых цветов

Изначально содержит три элемента цвета: красный(255, 0, 0), синий(0, 0, 255) и зеленый(0, 255, 0). При нажатии левой кнопкой мыши на элемент цвета, он добавляется на рабочую область. По нажатию кнопки «D» или «Пробел» происходит удаление всех добавленных в палитру элементов цвета, кроме изначальных.

Элемент цвета

Отображается в виде примитива — круга, чей цвет выставляется согласно указанному для элемента RGB значению.

При добавлении в рабочую область, элемент размещается по центру. В зависимости от координаты x(горизонтальной) регулируется оттенок цвета. Оттенок соответствует изначальному цвету (крайнее правое положение) или соответствует черному цвету (крайнее левое положение).

Пример:

Зеленый элемент с цветом (0,255,0):

- В крайнем правом положении значение оттенка будет (0, 255, 0).
- В крайнем левом положении значение оттенка будет (0, 0, 0).
- При положении посередине рабочей области значение оттенка будет (0, 122, 0), округление в меньшую сторону.

С элементом можно взаимодействовать:

- В рабочей области:
 - Перетаскивание элемента нажатием левой кнопки мыши;
 - Удаление элемента нажатием правой кнопки мыши.
- В палитре:
 - Добавление элемента в рабочую область по нажатию левой кнопки мыши.



Взаимодействие с элементом вызывает обновление цветового дисплея. При обновлении дисплея итоговый цвет высчитывается, как среднее арифметическое значений каналов RGB всех оттенков, кроме каналов равных нулю.

Пример:

На рабочей области расположены три элемента цвета:

- Синий (0, 0, 255) с координатами (0, 0).
- Красный (255, 0, 0) с координатами (80, 40).
- Синий (0, 0, 255) с координатами (400, 30).

Рабочая область имеет ширину в 400 пикселей.

Тогда итоговый цвет дисплея будет иметь цвет:

- $R = (255 * 80 / 400) / 1$
- $G = 0$
- $B = (255 * 400 / 400 + 255 * 0 / 400) / 2$

Что в итоге даст RGB(51, 0, 122).

Рабочая область

Содержит все добавленные из палитры элементы цвета, с которыми пользователь может взаимодействовать, согласно указанному функционалу. С помощью левой кнопки мыши можно перетаскивать добавленные элементы по рабочему полю. По нажатию правой кнопки мыши, элемент цвета удаляется из рабочего поля.

Добавление, удаление и взаимодействие с элементами цвета должно вызывать обновление цветового дисплея.

Цветовой дисплей

Принимает цвет, в зависимости от оттенков, размещенных на рабочей области элементов цвета. Цвет вычисляется как среднее арифметическое значение каналов RGB, кроме равных 0. При нажатии левой кнопки мыши на дисплей новый элемент цвета добавляется в палитру, его цвет должен совпадать с текущим цветом дисплея.



Требования к проекту

В рамках предоставленного задания вам необходимо продемонстрировать владение следующими принципами ООП:

- Четыре основные сущности должны быть представлены в виде классов;
- Эти классы должны иметь общий базовый класс и переопределять наследуемый метод;
- Продемонстрируйте преобразование типов;
- Продемонстрируйте разграничение доступа, используя инкапсуляцию.

Пример:

Созданы 4 класса: элемент цвета, палитра, рабочая область и цветовой дисплей.

Все эти классы наследуют базовый класс `UIElement`, от которого наследуют и переопределяют метод `Draw` отрисовки элемента в окне приложения.

В программе объявлена коллекция, хранящая экземпляры класса `UIElement`, в которой хранятся все экземпляры классов основных сущностей. Отрисовка всех элементов интерфейса происходит в цикле по данной коллекции.

Класс элементов цвета имеет закрытое поле `Coords`, обозначающее расположение данного элемента на экране. Изменение данного поля происходит через вызов открытого метода `Move`.

В рамках предоставленного задания вам необходимо продемонстрировать владение механизмом обработки исключений в ходе работы программы.

Пример:

При обновлении цветового дисплея поместите в блок обработки исключений обход коллекции элементов цвета рабочей области для проверки инициализации коллекции.

В рамках предоставленного задания вам необходимо продемонстрировать владение следующими навыками оформления кода:

- Все переменные, классы и методы должны иметь говорящие имена;
- Весь код должен быть написан в едином стиле (`CamelCase`, `snake_case` и т.д.);
- Классы и методы должны иметь комментарий.