

1-й этап разработки курсовой работы — анализ предметной области.

По результатам этого анализа оформляется первый раздел основной части, включающий краткое описание задачи и предметной области (объекты и их поведение, взаимодействие), требования к программе.

С точки зрения ОО подхода *объект представляет собой конкретный опознаваемый предмет, единицу или сущность (реальную или абстрактную), имеющую четко определенное функциональное назначение в данной предметной области.*

Объекты соответствуют существительным в описании предметной области, не глаголам или прилагательным. Глаголы превращаются в операции над объектами, а прилагательные - в свойства объектов или параметры операций.

Объект может иметь одну характеристику и тогда он может быть представлен простой переменной, но чаще характеристик несколько и нужно объединить их в struct или class.

Объекты могут быть составными и включать другие объекты. Например, палитра для выбора цвета рисования состоит из 16 кнопок.

Если мы не взаимодействуем с элементами составного отдельно, то объект можно не представлять в виде составного. Например, поле для вывода координат включает вывод координаты X и координаты Y, но мы всегда выводим их одновременно, поэтому это поле можно рассматривать как простой объект.

В постановку задачи входит только первая часть, без рекомендаций по реализации, интерфейса классов и пользовательского интерфейса, по развитию или упрощению темы.

Ниже рассмотрены два примера.

## Пример 1. Растровый графический редактор

### 1 Постановка задачи

Необходимо разработать растровый графический редактор Paint со следующими возможностями:

- рисование точек (свободное рисование);
- рисование линий;
- рисование прямоугольников (заполненных и нет);
- выбор цветов рисования и заполнения из 16 или более;
- чтение и запись рисунка в стандартном (.BMP, .PCX, .JPG) или собственном формате.

При движении мыши высвечивать координаты.

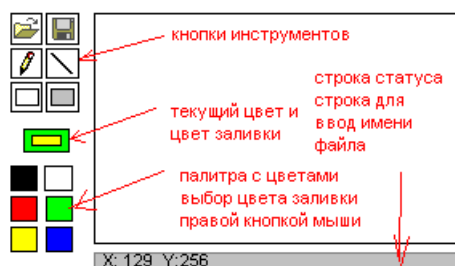


Рисунок 1 – Примерный интерфейс программы

Анализ предметной области выявляет следующие объекты:

- кнопки управления, имеющие квадратную форму и отличающиеся изображением или цветом;
- объект, отображающий текущие цвета рисования;
- объект, отображающий текущие координаты мыши и вводимое имя файла при загрузке и сохранении;
- поле для рисования, реакция которого на движение мыши зависит от текущего выбранного цвета и инструмента.

Кнопки управления по поведению можно разделить на 3 группы, различающиеся по поведению:

- 16 кнопок выбора цвета, изменяющие состояние объекта, отображающего текущие цвета рисования
- 4 кнопки выбора инструмента;
- 2 кнопки загрузки и сохранения рисунка.

Требования к программе:

- графический интерфейс;
- работа с мышью.

## Пример 2. Библиотека классов для работы с форматом JSON

### 1 Постановка задачи

Необходимо разработать библиотеку классов для работы с информацией в формате JSON.

Элементарными видами значений в JSON являются константы true, false и null (отсутствие значения), вещественные числа, строки, которые записываются как в языке C. Из них строятся массивы — последовательность значений, которые записываются в квадратных скобках через запятую. Нумерация элементов в массиве начинается с 0. Ассоциативные массивы записываются в фигурных скобках как набор пар *ключ:значение* через запятую. В качестве ключа используются только строки, значение может быть любого вида.

Пример: { "name": "John", "student": true, "age": 27, "bag": ["book", 7.5], "spouse": null }

Необходимые операции:

- чтение из потока;
- запись в поток;
- определить тип значения;
- проверка наличия элемента ассоциативного массива по ключу;
- удаление элементов из ассоциативного массива по ключу;
- получение длины массива
- добавление и удаление значения в массив (как в стеке)
- обращение к элементу массива по индексу и ассоциативного массива по ключу, с автоматическим добавлением как null при отсутствии;
- преобразование элементарных значений в стандартные типы данных (bool, double, string).

Анализ предметной области выявляет, что объекты могут хранить любое из 6 видов значений (null, булевское, число, строку, массив, ассоциативный массив). Но поведение объектов должно меняться в зависимости от вида значения. Некоторые операции можно применять к значению любого вида: чтение и запись в поток, определение типа значения. Другие операции только для массивов и ассоциативных массивов. Операции преобразования - только для элементарных значений.