Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

|  |
| --- |
| **Отчет** |
| по контрольной работе по дисциплине «Программирование» |
| наименование темы  Идиомы программирования |
|  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | ЭВМб-23-3 |  |  |  | Распутин Г. М. |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | Столбов А. Б |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2024 г

**Данная работа выполнена в команде из 3-х человек:**

Мельников Сергей Викторович, ЭВМб-23-3

Распутин Григорий Максимович, ЭВМб-23-3

Бобырев Данил Дмитриевич, ЭВМб-23-3

# Задание 1.

|  |  |  |
| --- | --- | --- |
| **Описание идиомы** | **Номер строк кода** | **Комментарий** |
| [Включение стандартных библиотек](https://hyperpolyglot.org/cpp#implicit-prologue-note) | 1-2 |  |
| [Директивы пространства имён](https://hyperpolyglot.org/cpp#import-namespace) | 3 |  |
| [Объявление функции](https://hyperpolyglot.org/cpp#decl-func) [типа int](https://hyperpolyglot.org/cpp#int-type-note) | 5, 49 |  |
| [Объявление функции](https://hyperpolyglot.org/cpp#decl-func) [типа void](https://hyperpolyglot.org/cpp#no-retval-note) | 23 |  |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа bool](https://hyperpolyglot.org/cpp#boolean-type-note) | 33, 61 |  |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа int](https://hyperpolyglot.org/cpp#int-type-note) | 6, 27, 51 |  |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа int\*\*](https://hyperpolyglot.org/cpp#allocate-heap-note) | 55 |  |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа string](https://hyperpolyglot.org/cpp#str-type) | 7 |  |
| [Стандартный вывод](https://hyperpolyglot.org/cpp#hello-world-note) | 8, 19, 37-38, 52, 57, 100, 103, 105, 107, 108, 110, 112, 114 |  |
| [Стандартный ввод](https://hyperpolyglot.org/cpp#read-line-stdin-note) | 28 |  |
| [Цикл while](https://hyperpolyglot.org/cpp#while) | 10 |  |
| [Цикл for](https://hyperpolyglot.org/cpp#for-note) | 24, 26, 34, 35, 62, 71, 72, 85-87, 101, 102, 118 |  |
| [Строковой поток](https://hyperpolyglot.org/cpp#fmt-str-note) | 11 |  |
| [Условный оператор с одном блоком](https://hyperpolyglot.org/cpp#if-note) | 12-14, 36, 44, 63, 73, 78, 88, 93, 96 |  |
| [Оператор возврата значения из функции](https://hyperpolyglot.org/cpp#retval) | 15, 123 |  |
| [Оператор присваивания](https://hyperpolyglot.org/cpp#assignment-note) | 6, 25, 29, 33, 40, 53, 64, 74, 83, 89 |  |
| [Тернарный оператор](https://hyperpolyglot.org/cpp#if-note) | 108-116 |  |
| [Оператор возвращения памяти в кучу(delete)](https://hyperpolyglot.org/cpp#free-heap-note) | 119, 121 |  |
| [Оператор выхода из цикла(break)](https://hyperpolyglot.org/cpp#break) | 41, 45, 65, 75, 79, 90, 94, 97, |  |
| [Вызов функции](https://hyperpolyglot.org/cpp#call-func-note) | 53, 59, |  |
| [Установка локали](https://en.cppreference.com/w/cpp/locale/setlocale) | 50 |  |

Код лабораторной 3:

|  |  |
| --- | --- |
| 1. | #include <iostream> |
| 2. | #include <sstream> |
| 3. | using namespace std; |
| 4. |  |
| 5. | int intInputFromConsole(string mess, string err, int min, int max) { |
| 6. | int curNumber = 0; |
| 7. | string line; |
| 8. | cout << mess; |
| 9. |  |
| 10. | while (getline(cin, line)) { |
| 11. | stringstream ss(line); |
| 12. | if (ss >> curNumber) { |
| 13. | if (ss.eof()) { |
| 14. | if (curNumber >= min && curNumber <= max) { |
| 15. | return curNumber; |
| 16. | } |
| 17. | } |
| 18. | } |
| 19. | cout << err << "\n"; |
| 20. | } |
| 21. | } |
| 22. |  |
| 23. | void checkerMatrix(int\*\* relation, int n) { |
| 24. | for (int i = 0; i < n; i++) { |
| 25. | relation[i] = new int[n]; |
| 26. | for (int j = 0; j < n; j++) { |
| 27. | int a; |
| 28. | cin >> a; |
| 29. | relation[i][j] = a; |
| 30. | } |
| 31. | } |
| 32. |  |
| 33. | bool checkchek = true; //Переменная для того, чтобы выйти из вложенного цикла |
| 34. | for (int i = 0; i < n; i++) { |
| 35. | for (int j = 0; j < n; j++) { |
| 36. | if (relation[i][j] != 1 && relation[i][j] != 0) { |
| 37. | cout << "Ошибка ввода матрицы, попробуйте снова\n"; |
| 38. | cout << "Неверное число: " << relation[i][j] << '\n'; |
| 39. | checkerMatrix(relation, n); |
| 40. | checkchek = false; |
| 41. | break; |
| 42. | } |
| 43. | } |
| 44. | if (checkchek == false) |
| 45. | break; |
| 46. | } |
| 47. | } |
| 48. |  |
| 49. | int main() { |
| 50. | setlocale(LC\_ALL, "ru"); |
| 51. | int n; |
| 52. | cout << "Элементов в множесетве будет: "; |
| 53. | n = intInputFromConsole("", "erorr", 1, 10); |
| 54. |  |
| 55. | int\*\* relation = new int\* [n]; |
| 56. |  |
| 57. | cout << "Введите бинарные отношения для матрицы (" << n << "x" << n << "):" |
| 58. | << "\n"; |
| 59. | checkerMatrix(relation, n); |
| 60. |  |
| 61. | bool reflex = true; |
| 62. | for (int i = 0; i < n; i++) { |
| 63. | if (relation[i][i] != 1) { |
| 64. | reflex = false; |
| 65. | break; |
| 66. | } |
| 67. | } |
| 68. |  |
| 69. | //проверка на симметричность |
| 70. | bool symetr = true; |
| 71. | for (int i = 0; i < n; i++) { |
| 72. | for (int j = 0; i < n; i++) { |
| 73. | if (relation[i][j] != relation[j][i]) { |
| 74. | symetr = false; |
| 75. | break; |
| 76. | } |
| 77. | } |
| 78. | if (symetr == false) |
| 79. | break; |
| 80. | } |
| 81. |  |
| 82. | //проверка на транзитивность |
| 83. | bool tran = true; |
| 84. |  |
| 85. | for (int i = 0; i < n; i++) { |
| 86. | for (int j = 0; j < n; j++) { |
| 87. | for (int k = 0; k < n; k++) { |
| 88. | if (relation[i][j] == 1 && relation[j][k] == 1 && relation[i][k] != 1) { |
| 89. | tran = false; |
| 90. | break; |
| 91. | } |
| 92. | } |
| 93. | if (tran == false) |
| 94. | break; |
| 95. | } |
| 96. | if (tran == false) |
| 97. | break; |
| 98. | } |
| 99. |  |
| 100. | cout << "Матрица бинарных отношений: \n"; |
| 101. | for (int i = 0; i < n; i++) { |
| 102. | for (int j = 0; j < n; j++) { |
| 103. | cout << relation[i][j] << " "; |
| 104. | } |
| 105. | cout << "\n"; |
| 106. | } |
| 107. | cout << "\n\n\n"; |
| 108. | cout << (reflex ? "бинарное отношение рефлексивно!!!\n" |
| 109. | : "бинарное отношение не рефлексивно.\n"); |
| 110. | cout << (symetr ? "бинарное отношение симметрично!!!\n" |
| 111. | : "бинарное отношение не симметрично.\n"); |
| 112. | cout << (tran ? "бинарное отношение транзитивно!!!\n" |
| 113. | : "бинарное отношение не транзитивно.\n"); |
| 114. | cout << ((tran && symetr && reflex) |
| 115. | ? "бинарное отношение эквивалентно!!!\n" |
| 116. | : "отношение белок не эквивалентно.\n"); |
| 117. |  |
| 118. | for (int i = 0; i < n; i++) { |
| 119. | delete[] relation[i]; |
| 120. | } |
| 121. | delete[] relation; |
| 122. |  |
| 123. | return 0; |
| 124. | } |

# Задание 2.

**Аннотируемая библиотека** – OpenCV для C++

**Описание библиотеки:**

OpenCV (Open Source Computer Vision Library) - это библиотека с открытым исходным кодом, которая предоставляет широкий спектр функций и включает в себя несколько сотен алгоритмов компьютерного зрения. Она написана на C++ и может быть использована в различных языках программирования, включая Python, Java и C#.

**Описание темы:**

Компьютерное зрение - это перспективная область, которая всегда будет играть важную роль в нашей жизни. Она имеет потенциал для решения многих проблем, стоящих перед человечеством, и создания новых возможностей для развития.

Существует множество способов получения цифровых изображений из реального мира: цифровые камеры, сканеры, компьютерная томография, магнитно-резонансная томография и другие. В каждом случае то, что мы (люди) видим, - это изображения. Однако, преобразуя их в цифровые устройства, мы записываем числовые значения для каждой из точек изображения.

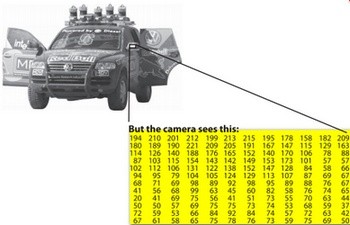


Рисунок 1 – числовое изображение точки

Например, на изображении (рис. 1) видно, что зеркало автомобиля - это не что иное, как матрица, содержащая все значения интенсивности точек-пикселей. Способ получения и хранения значений пикселей может варьироваться в зависимости от наших потребностей, но в конечном итоге все изображения в компьютерном мире могут быть сведены к числовым матрицам и другой информации, описывающей саму матрицу. OpenCV - это библиотека компьютерного зрения, основной задачей которой является обработка и манипулирование этой информацией.

**Установка библиотеки:**

**На Windows:**

1. Скачать архив с библиотекой OpenCV с официального сайта: <https://opencv.org/releases/>.
2. Распаковать архив в папку с вашим проектом(рис. 2).

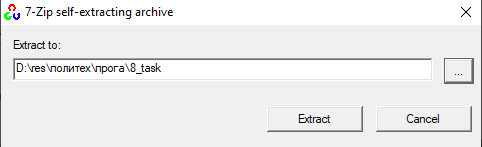


Рисунок 2 – распаковка архива

1. Изменить настройки системы для работы с OpenCV (рис. 3)
   1. При помощи поиска найти “Изменение системных переменных среды”
   2. Нажать кнопку “Переменные среды”
   3. Выбрать Path и нажать изменить

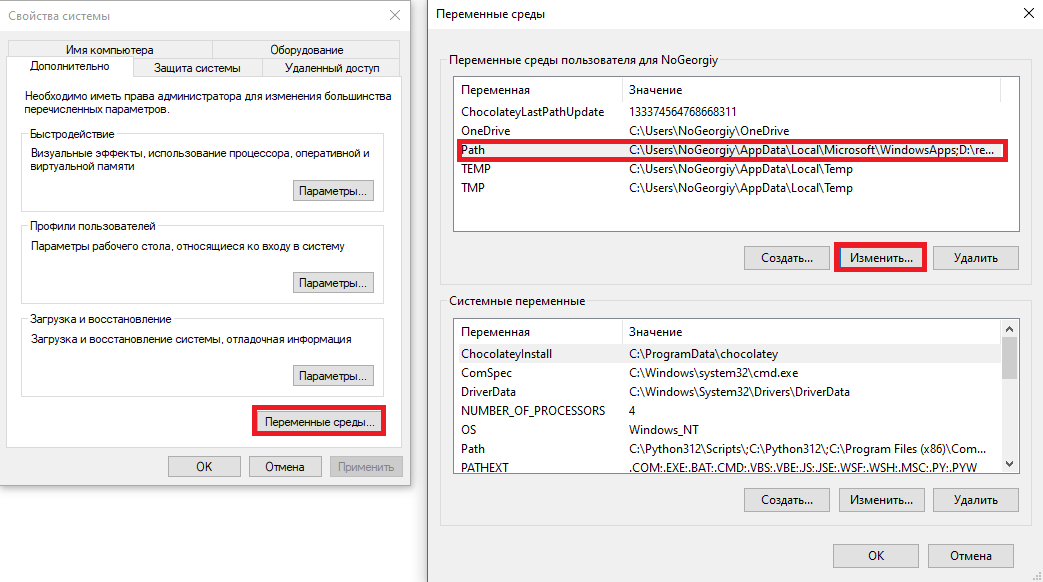


Рисунок 3 – изменение настроек системы

* 1. Создаём два новых пути - **<ваш-путь>\opencv\build** и

**<ваш-путь>\opencv\build\x64\vc16\bin** (рис. 4)

c. Ок

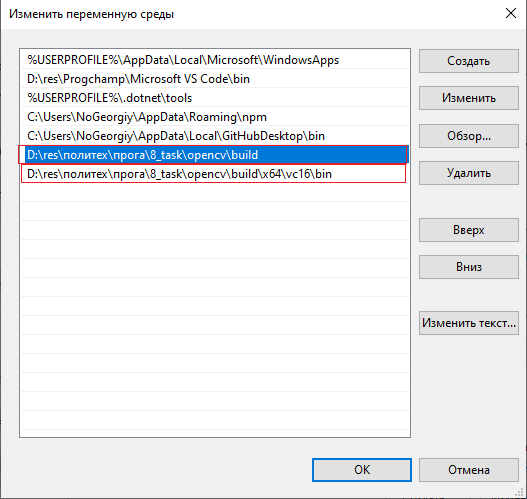


Рисунок 4 – создание нового пути

1. Настроить среду visual studio
   1. Открыть окно свойств проекта (рис. 5)

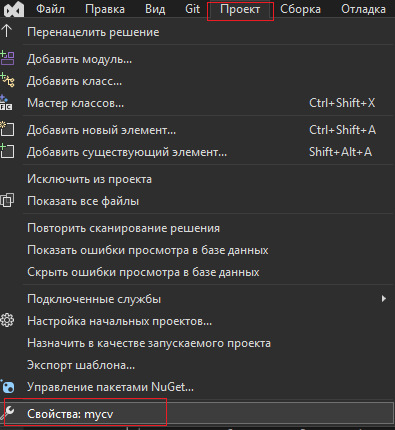


Рисунок 5 – настройка VS

* 1. Во вкладке Каталоги VC++ -> **Включаемые каталоги** изменить значение на **<ваш-путь>\opencv\build\include (рис. 6)**

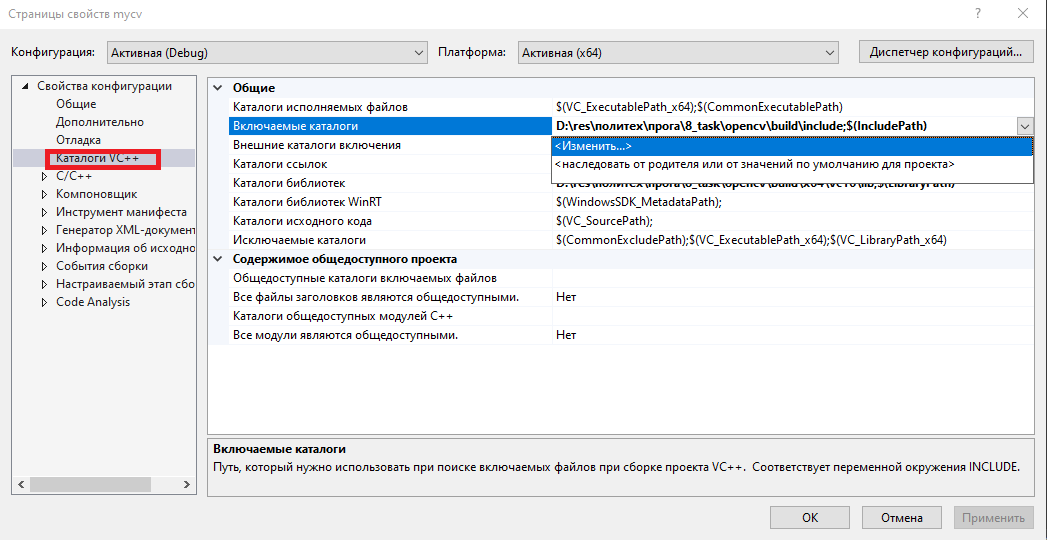


Рисунок 6 – смена включаемого каталога

* 1. В этой же вкладке изменить значение **Каталог библиотек** на **<ваш-путь>\ opencv\build\x64\vc16\lib (рис. 7)**

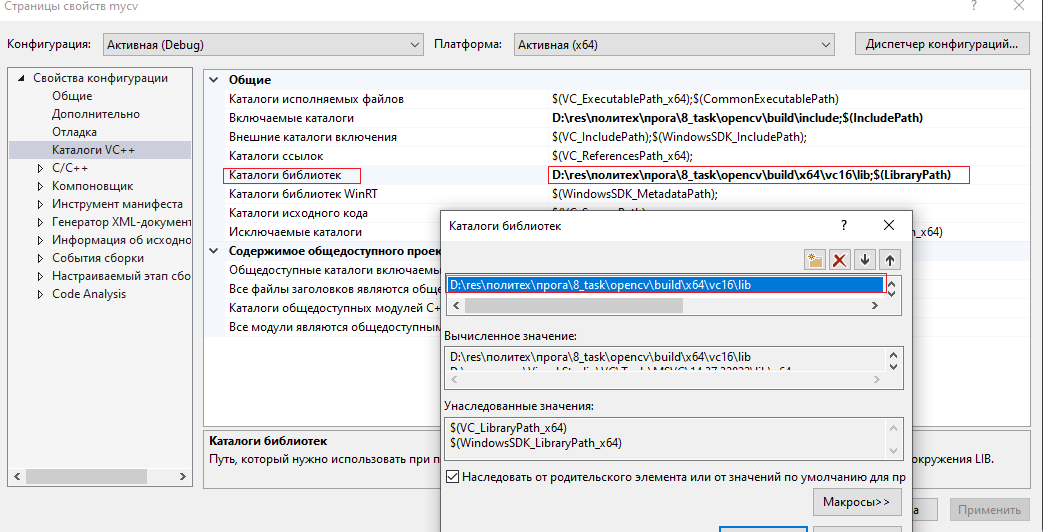


Рисунок 7 – смена пути в каталоге библиотек

* 1. Во вкладке Отладка изменить значение **Окружения** на **PATH=<ваш-путь>** \**opencv\build\x64\vc16\bin (рис. 8)**

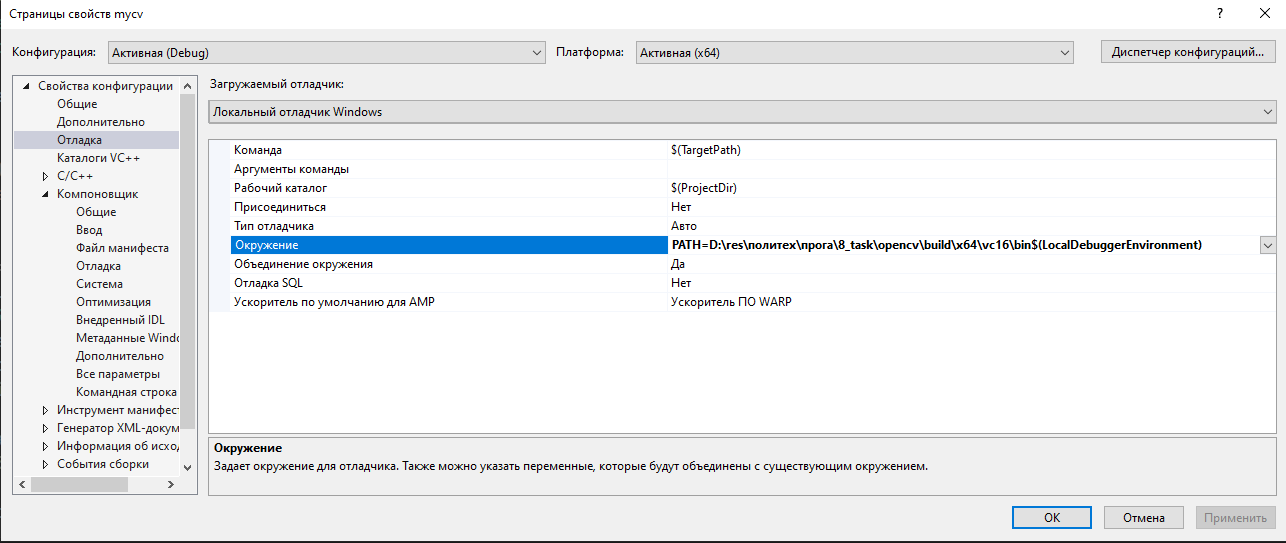


Рисунок 8 – смена значения окружения

* 1. Во вкладке Компоновщик->Общее изменить значение “Включить инкрементную компоновку” на **нет (рис. 9)**

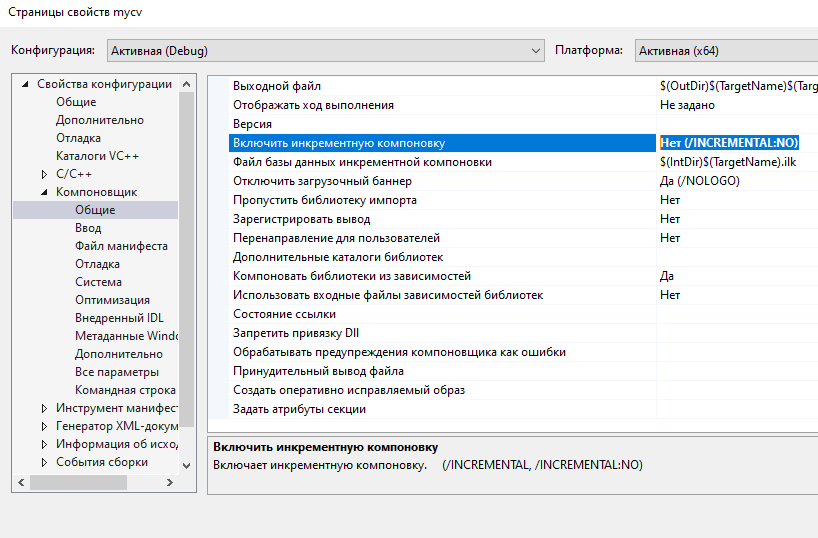


Рисунок 9 – отключение инкрементной компоновки

* 1. Во вкладке Компоновщик->Ввод изменить значение “Дополнительные зависимости” на название файла, который находится в папке **<ваш-путь>** \**opencv\build\x64\vc16\lib.** У файла должно быть расширение **.lib** и он должен заканчиваться на **d. (рис. 10)**

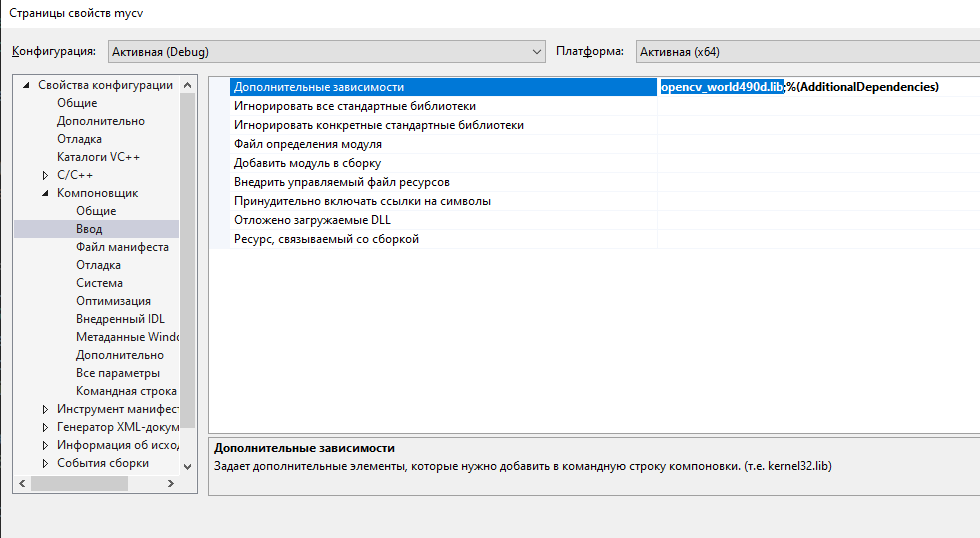


Рисунок 10 – смена значения дополнительных зависимостей

1. Библиотека успешно установлена и готова к работе.

**На MacOS:**

1. Скачать установщик пакетов HomeBrew(рис. 11)

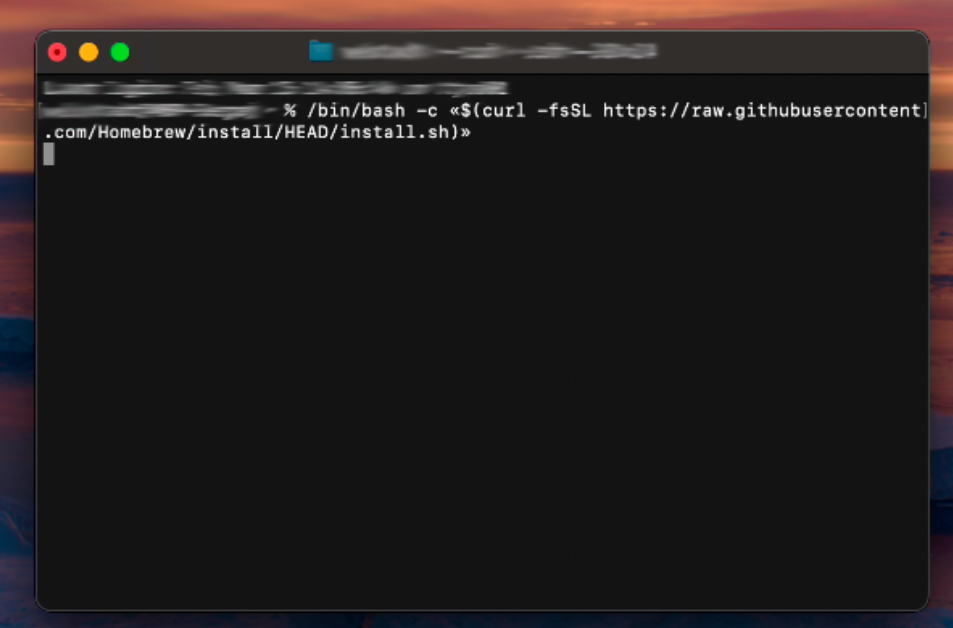


Рисунок 11 – установка HomeBrew через терминал

2. Установить пакет OpenCV с помощью HomeBrew, введя команду «brew install opencv» (рис. 12)

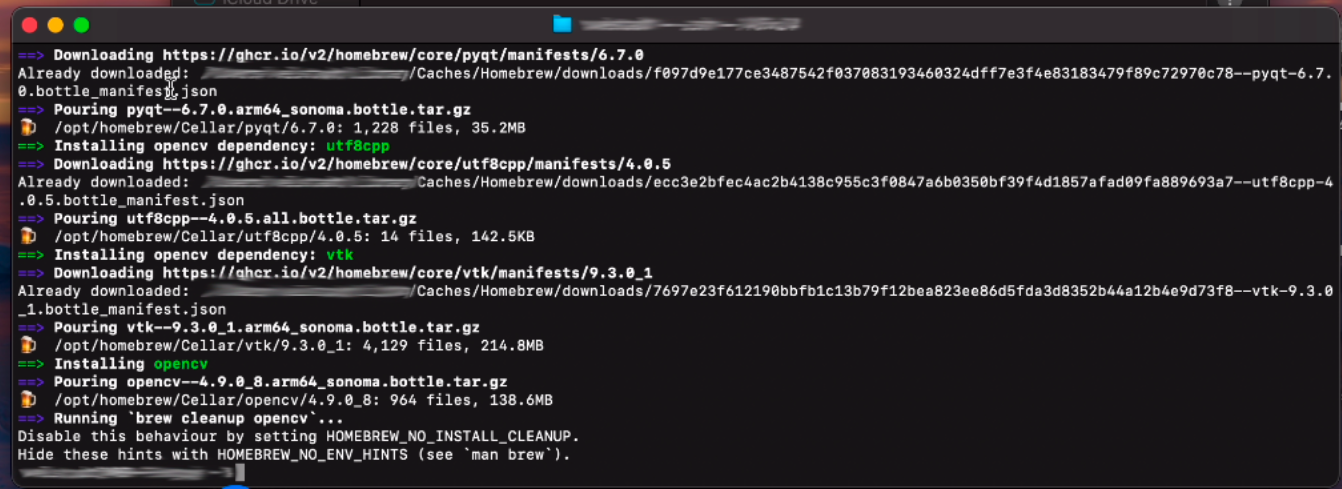


Рисунок 12 – Установки библиотеки OpenCV через терминал

3. Создать новый проект в среде разработки Xcode (рис. 13,14)



Рисунок 13 – запущенная среда разработки Xcode

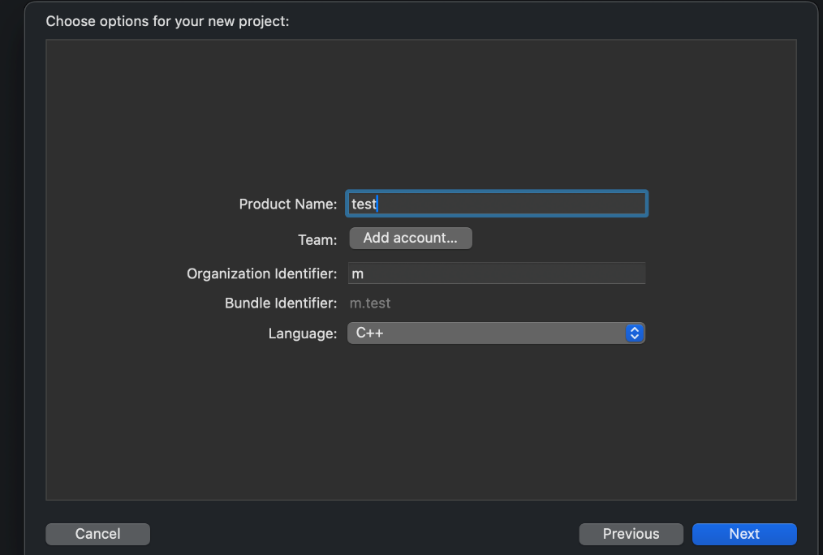


Рисунок 14 – создание консольного файла в Xcode

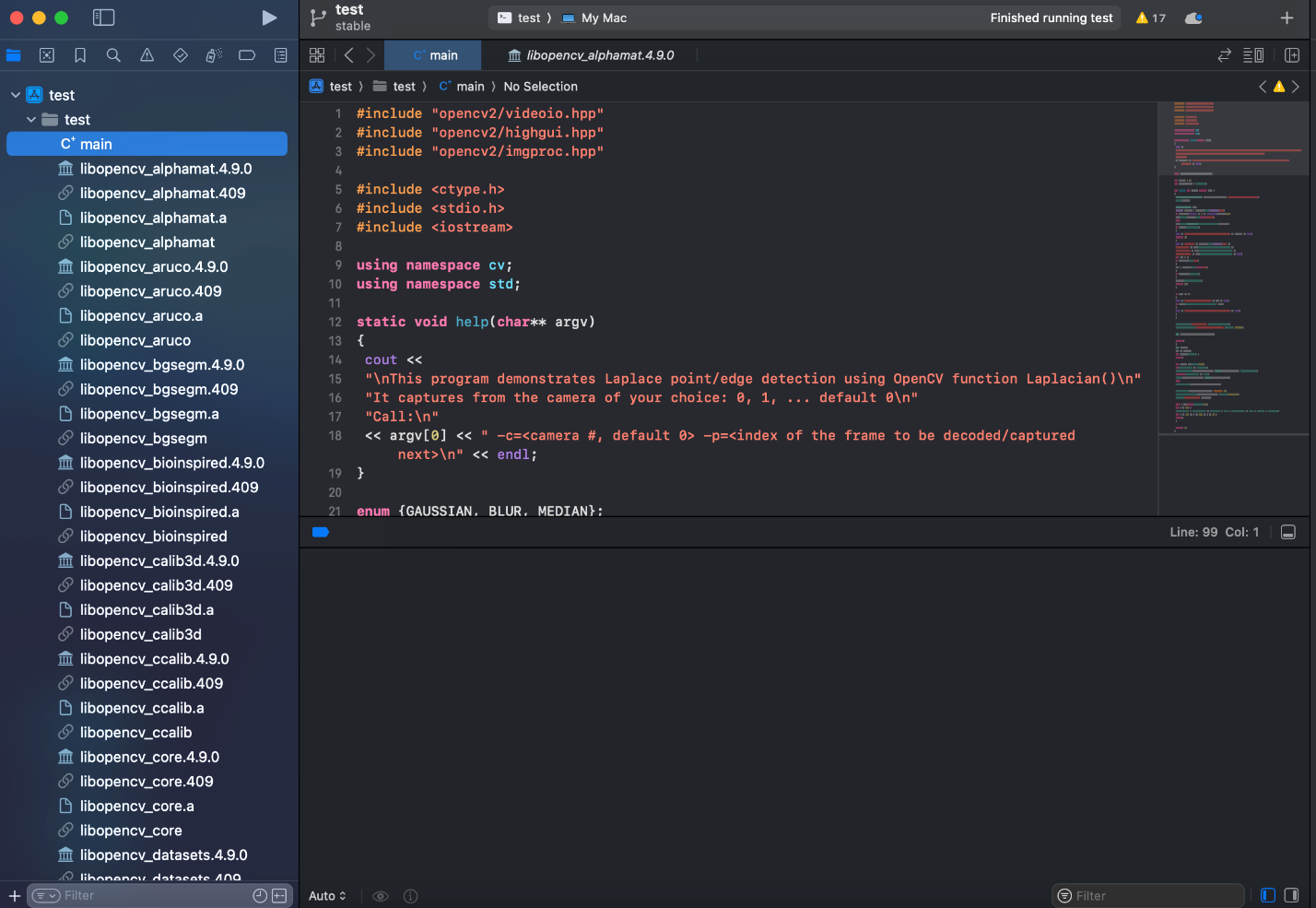
4. Загрузить в проект необходимые модули из папки OpenCV, просто перетащив их из finder в окно Xcode (рис. 15)

Рисунок 15 – подключение модулей OpenCV к проекту

5. Библиотека успешно установлена и готова к работе

## **Аннотирование:**

## **Аннотирование функции** [**Colormap.cpp**](https://github.com/opencv/opencv/blob/4.x/modules/imgproc/src/colormap.cpp#L28)

|  |  |  |
| --- | --- | --- |
| **Описание идиомы** | **Номер строк кода** | **Комментарий** |
| [Включение стандартных библиотек](https://hyperpolyglot.org/cpp#implicit-prologue-note) | 18, 19 |  |
| [Директива #ifndef](https://learn.microsoft.com/ru-ru/cpp/preprocessor/hash-ifdef-and-hash-ifndef-directives-c-cpp?view=msvc-170) | 21 |  |
| [Директива #pragma](https://learn.microsoft.com/ru-ru/cpp/preprocessor/pragma-directives-and-the-pragma-keyword?view=msvc-170) | 22 |  |
| [Директива #endif](https://learn.microsoft.com/ru-ru/windows/win32/menurc/-endif) | 23 |  |
| [Директивы пространства имён](https://hyperpolyglot.org/cpp#import-namespace) | 25, 133 |  |
| [Объявление функции](https://hyperpolyglot.org/cpp#decl-func) | 28, 40, 55, 63, 75, 111, 177, 181, 185, … , 720, 723, 728, 790, 805, 839 | 28 - Эта функция генерирует вектор из n точек на одинаковом расстоянии между x0 и x1.  40, 55 - сортировка строк матрицы по индексам.  63 - Эта функция сортирует элементы матрицы и возвращает индексы отсортированных элементов.  75, 111 - Эта функция выполняет линейную интерполяцию в одномерной таблице.  177 – Конструктор по умолчанию. Инициализирует цветовую карту с 256 цветами.  181 – Конструктор, который создаёт карту цветов с заданным количеством цветов.  185 - Метод init инициализирует данные карты цветов.  177,181,185 повторяется для каждого класса цветовой карты.  720 – Конструктор с пользовательской цветовой картой  728 - Эта функция перегружает оператор() класса ColorMap.  790 - Эта функция создает линейную карту цветов на основе предоставленных входных данных.  805 - Эта функция применяет предопределенную карту цветов к входному изображению.  839 - Эта функция применяет определенную пользователем цветовую карту к исходному изображению. |
| [Объявление переменной типа](https://hyperpolyglot.org/cpp#local-var-note) [Mat](https://docs.opencv.org/4.x/d3/d63/classcv_1_1Mat.html) | 28, 30, 44, 47, 49, …, 139 | Создает переменную типа mat. Она может хранить в себе векторы или матрицы, а также изображения |
| [Объявление переменной типа](https://hyperpolyglot.org/cpp#local-var-note) [Float](https://hyperpolyglot.org/cpp#float-type-note) | 31, 164, 186, … , 689 |  |
| [GetMat](https://docs.opencv.org/3.4/d7/d45/classcv_1_1UMat.html#a96b462ce3e2edc6344bcc72ee4b8f5cb) | 42, 44, 45, 47, 65, 114-116, 734, 752, 845 | метод, который позволяет получить матрицу из объекта изображения. |
| [at](https://hyperpolyglot.org/cpp#resizable-array-lookup-note) | 33, 89, 91, 96, 103-106 |  |
| [Цикл for](https://hyperpolyglot.org/cpp#for-note) | 32, 48, 85, 147, 683, 766, 769, 779, 782 |  |
| [Инициализация вектора](https://hyperpolyglot.org/cpp#decl-resizable-array-note) | 45, 79 |  |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа int](https://hyperpolyglot.org/cpp#int-type-note) | 68, 77, … |  |
| [template](https://hyperpolyglot.org/cpp#define-generic) | 74 |  |
| [Цикл while](https://hyperpolyglot.org/cpp#while) | 94 |  |
| [Оператор switch](https://hyperpolyglot.org/cpp#switch-note) | 121 |  |
| [Class](https://hyperpolyglot.org/cpp#define-class) | 136, 175, 199, … , 713 |  |
| [Определение метода класса](https://hyperpolyglot.org/cpp#def-static-class-method-note) | 138, 3, 141, 175, … , 714 |  |
| [this](https://hyperpolyglot.org/cpp#overload-op-note) | 8, 10, 13, 22, …, 721 |  |
| [CV\_Error](https://docs.opencv.org/3.4/d1/d0d/namespacecv_1_1Error.html) | 43, 67, 130, … , 844 |  |
| [merge](https://en.cppreference.com/w/cpp/algorithm/merge) | 798 |  |
| [delete](https://hyperpolyglot.org/cpp#free-heap-note) | 836 |  |
| [linspace](https://numpy.org/doc/stable/reference/generated/numpy.linspace.html) | 28, 158, 165, 189, … , 701 | создает последовательность чисел, которые равномерно распределяются между начальной и конечной точкой. |
| [Условный оператор if](https://hyperpolyglot.org/cpp#if-note) | 42, 66, 89, 91, … , 843 |  |
| [Else](https://hyperpolyglot.org/cpp#dangling-else-note) | 98, 748, 775 |  |
| [new](https://hyperpolyglot.org/cpp#dict-ctor-note) | 808-829 |  |

## **Аннотирование примеров использования функций библиотеки.**

## **falsecolor.cpp**

|  |  |  |
| --- | --- | --- |
| **Описание идиомы** | **Номер строк кода** | **Комментарий** |
| [Включение стандартных библиотек](https://hyperpolyglot.org/cpp#implicit-prologue-note) | 1-4 |  |
| [Директивы пространства имён](https://hyperpolyglot.org/cpp#import-namespace) | 6-7 | На 6 строке использовано пространство имён, чтобы избежать постоянного написания cv перед методами библиотеки |
| [Объявление enum](https://hyperpolyglot.org/cpp#enum) | 9 |  |
| [Объявление структуры](https://hyperpolyglot.org/cpp#struct-initialization) | 11 | определяет структуру для хранения индекса цветовой карты и изображения. |
| [Объявление глобальной переменной](https://hyperpolyglot.org/cpp#global-var-note) | 16, 17 | 16 – имя окна  17 – массив имен доступных цветовых карт |
| [Объявление функции](https://hyperpolyglot.org/cpp#decl-func) | 21, 40, 83 | 21 – Функция, вызываемая при каждом изменении трекбара  40 – Создаёт изображение различных форм  83 – вход в приложение |
| [Объявление переменной типа](https://hyperpolyglot.org/cpp#local-var-note) [Mat](https://docs.opencv.org/4.x/d3/d63/classcv_1_1Mat.html) | 13, 24, 28, 40, 42, 88 | Создает переменную типа mat. Она может хранить в себе векторы или матрицы, а также изображения |
| [Условный оператор if](https://hyperpolyglot.org/cpp#if-note) | 26, 90 |  |
| [else](https://hyperpolyglot.org/cpp#dangling-else-note) | 32, 92 |  |
| [randu](https://docs.opencv.org/4.x/d2/de8/group__core__array.html#ga1ba1026dca0807b27057ba6a49d258c0) | 29 | Генерирует одно равномерно распределенное случайное число или массив случайных чисел. |
| [applyColorMap](https://docs.opencv.org/4.x/d3/d50/group__imgproc__colormap.html#gadf478a5e5ff49d8aa24e726ea6f65d15) | 30, 33 | Применяет эквивалентную цветовую карту GNU Octave/MATLAB к данному изображению. |
| [putText](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga5126f47f883d730f633d74f07456c576) | 35 | Функция [cv::putText](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga5126f47f883d730f633d74f07456c576) отображает указанную текстовую строку на изображении. Символы, которые невозможно отобразить с использованием указанного шрифта, заменяются вопросительными знаками. |
| [imshow](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#ga453d42fe4cb60e5723281a89973ee563) | 36, 98 | Отображает изображение в указанном окне. |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа int](https://hyperpolyglot.org/cpp#int-type-note) | 43, 44, 50-52 |  |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа point](https://docs.opencv.org/4.x/db/d4e/classcv_1_1Point__.html) | 49 | Класс шаблона для 2D-точек, заданных их координатами x и y. |
| [line](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga7078a9fae8c7e7d13d24dac2520ae4a2) | 47 | Рисует линию от одной точки к другой |
| [RNG](https://docs.opencv.org/4.x/d2/d44/classcv_1_1Rect__.html) | 48 | Генератор 2D-прямоугольников |
| [Rect](https://docs.opencv.org/4.x/d2/d44/classcv_1_1Rect__.html) | 53 | Перемененная для хранения координат прямоугольника |
| [Цикл for](https://hyperpolyglot.org/cpp#for-note) | 46, 55 |  |
| [Оператор switch](https://hyperpolyglot.org/cpp#switch-note) | 58 |  |
| [Оператор выхода (break)](https://hyperpolyglot.org/cpp#break) | 63,70,77 |  |
| [uniform](https://docs.opencv.org/4.x/d1/dd6/classcv_1_1RNG.html#acde197860cea91e5aa896be8719457ae) | 57,60,61,65,66,67,72-75 | возвращает равномерно распределенное целое случайное число из диапазона [a,b) |
| [circle](https://docs.opencv.org/4.x/d9/db7/group__datasets__gr.html#gga610754124ced68d1f05760b5948fbb76a6f0d8b2d9e3e947b2a5c1eff9e81ee95) | 62 | Рисует круг |
| [rectangle](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga07d2f74cadcf8e305e810ce8eed13bc9) | 69 | Рисует прямоугольник |
| [ellipse](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga28b2267d35786f5f890ca167236cbc69) | 76 | Рисует эллипс |
| [Стандартный вывод](https://hyperpolyglot.org/cpp#hello-world-note) | 107, 85 |  |
| [imread](https://docs.opencv.org/4.x/d4/da8/group__imgcodecs.html#gab32ee19e22660912565f8140d0f675a8) | 91 | Загружает изображение из файла. |
| [createTrackbar](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#gaf78d2155d30b728fc413803745b67a9b) | 100 | Создает трекбар и прикрепляет его к указанному окну. |
| [setTrackbarMin](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#gabe26ffe8d2b60cc678895595a581b7aa) | 101-103 | Устанавливает минимальное положение трекбара. |
| [waitKey](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#ga5628525ad33f52eab17feebcfba38bd7) | 108 | Ожидает нажатия клавиши. |
| [Оператор возврата значения из функции](https://hyperpolyglot.org/cpp#retval) | 80, 109 |  |

1. #include "opencv2/imgproc.hpp"

2. #include "opencv2/imgcodecs.hpp"

3. #include "opencv2/highgui.hpp"

4. #include <iostream>

5.

6. using namespace cv;

7. using namespace std;

8.

9. enum MyShape { MyCIRCLE = 0, MyRECTANGLE, MyELLIPSE };

10.

11. struct ParamColorMap {

12. int iColormap;

13. Mat img;

14. };

15.

16. String winName = "False color";

17. static const String ColorMaps[] = { "Autumn", "Bone", "Jet", "Winter", "Rainbow", "Ocean", "Summer", "Spring",

18. "Cool", "HSV", "Pink", "Hot", "Parula", "Magma", "Inferno", "Plasma", "Viridis",

19. "Cividis", "Twilight", "Twilight Shifted", "Turbo", "Deep Green", "User defined (random)" };

20.

21. static void TrackColorMap(int x, void\* r)

22. {

23. ParamColorMap\* p = (ParamColorMap\*)r;

24. Mat dst;

25. p->iColormap = x;

26. if (x == COLORMAP\_DEEPGREEN + 1)

27. {

28. Mat lutRND(256, 1, CV\_8UC3);

29. randu(lutRND, Scalar(0, 0, 0), Scalar(255, 255, 255));

30. applyColorMap(p->img, dst, lutRND);

31. }

32. else

33. applyColorMap(p->img, dst, p->iColormap);

34.

35. putText(dst, "Colormap : " + ColorMaps[p->iColormap], Point(10, 20), FONT\_HERSHEY\_SIMPLEX, 0.8, Scalar(255, 255, 255), 2);

36. imshow(winName, dst);

37. }

38.

39.

40. static Mat DrawMyImage(int thickness, int nbShape)

41. {

42. Mat img = Mat::zeros(500, 256 \* thickness + 100, CV\_8UC1);

43. int offsetx = 50, offsety = 25;

44. int lineLength = 50;

45.

46. for (int i = 0; i < 256; i++)

47. line(img, Point(thickness \* i + offsetx, offsety), Point(thickness \* i + offsetx, offsety + lineLength), Scalar(i), thickness);

48. RNG r;

49. Point center;

50. int radius;

51. int width, height;

52. int angle;

53. Rect rc;

54.

55. for (int i = 1; i <= nbShape; i++)

56. {

57. int typeShape = r.uniform(MyCIRCLE, MyELLIPSE + 1);

58. switch (typeShape) {

59. case MyCIRCLE:

60. center = Point(r.uniform(offsetx, img.cols - offsetx), r.uniform(offsety + lineLength, img.rows - offsety));

61. radius = r.uniform(1, min(offsetx, offsety));

62. circle(img, center, radius, Scalar(i), -1);

63. break;

64. case MyRECTANGLE:

65. center = Point(r.uniform(offsetx, img.cols - offsetx), r.uniform(offsety + lineLength, img.rows - offsety));

66. width = r.uniform(1, min(offsetx, offsety));

67. height = r.uniform(1, min(offsetx, offsety));

68. rc = Rect(center - Point(width, height) / 2, center + Point(width, height) / 2);

69. rectangle(img, rc, Scalar(i), -1);

70. break;

71. case MyELLIPSE:

72. center = Point(r.uniform(offsetx, img.cols - offsetx), r.uniform(offsety + lineLength, img.rows - offsety));

73. width = r.uniform(1, min(offsetx, offsety));

74. height = r.uniform(1, min(offsetx, offsety));

75. angle = r.uniform(0, 180);

76. ellipse(img, center, Size(width / 2, height / 2), angle, 0, 360, Scalar(i), -1);

77. break;

78. }

79. }

80. return img;

81. }

82.

83. int main(int argc, char\*\* argv)

84. {

85. cout << "This program demonstrates the use of applyColorMap function.\n\n";

86.

87. ParamColorMap p;

88. Mat img;

89.

90. if (argc > 1)

91. img = imread(samples::findFile(argv[1]), IMREAD\_GRAYSCALE);

92. else

93. img = DrawMyImage(2, 256);

94.

95. p.img = img;

96. p.iColormap = 0;

97.

98. imshow("Gray image", img);

99. namedWindow(winName);

100. createTrackbar("colormap", winName, NULL, COLORMAP\_DEEPGREEN + 1, TrackColorMap, (void\*)&p);

101. setTrackbarMin("colormap", winName, COLORMAP\_AUTUMN);

102. setTrackbarMax("colormap", winName, COLORMAP\_DEEPGREEN + 1);

103. setTrackbarPos("colormap", winName, COLORMAP\_AUTUMN);

104.

105. TrackColorMap(0, (void\*)&p);

106.

107. cout << "Press a key to exit" << endl;

108. waitKey(0);

109. return 0;

110. }

Пример работы приложения(рис. 16-18):

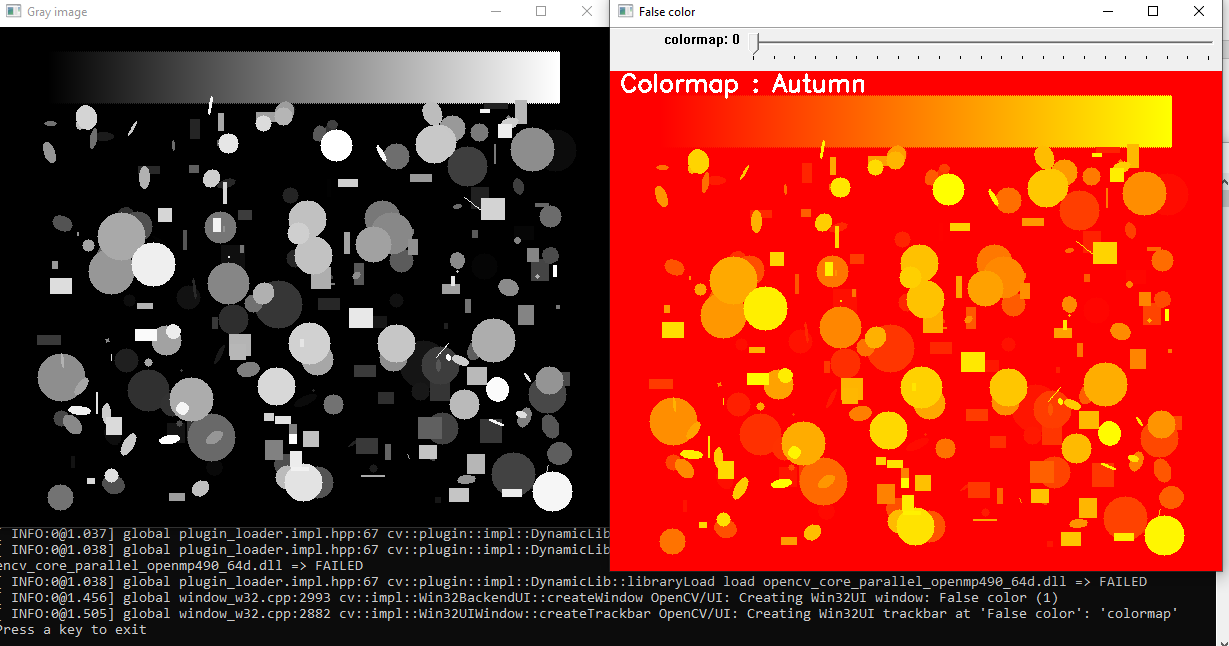


Рисунок 16 – Цветовая карта - autumn

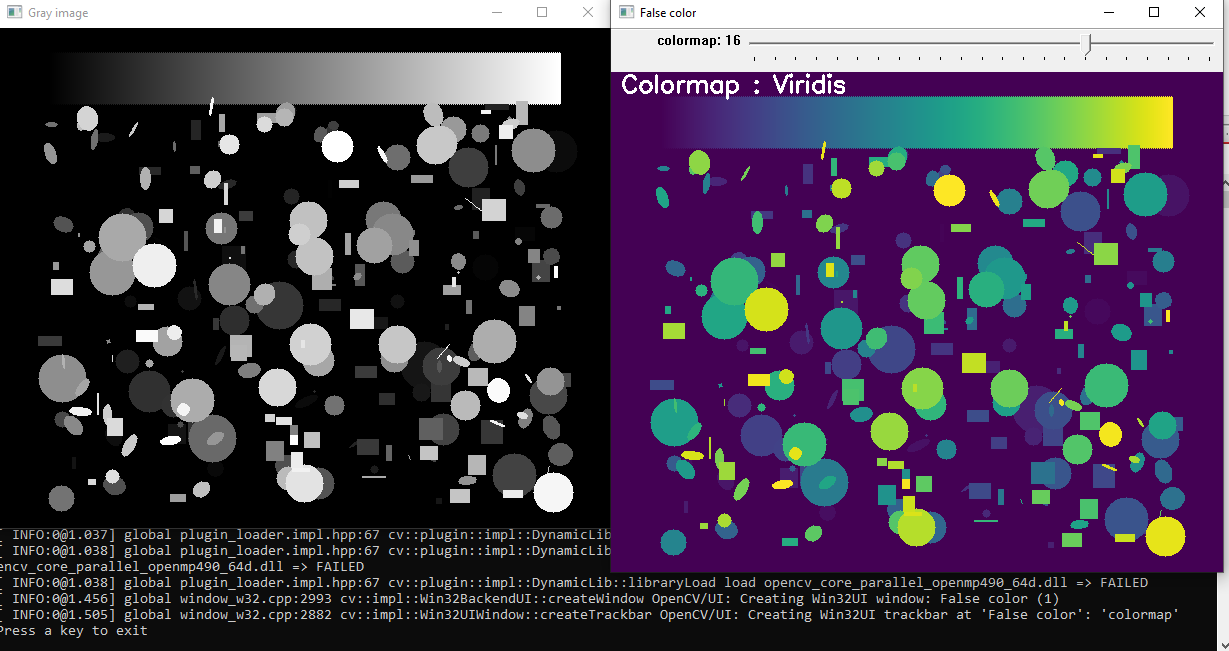


Рисунок 17 – Цветовая карта - Viridis

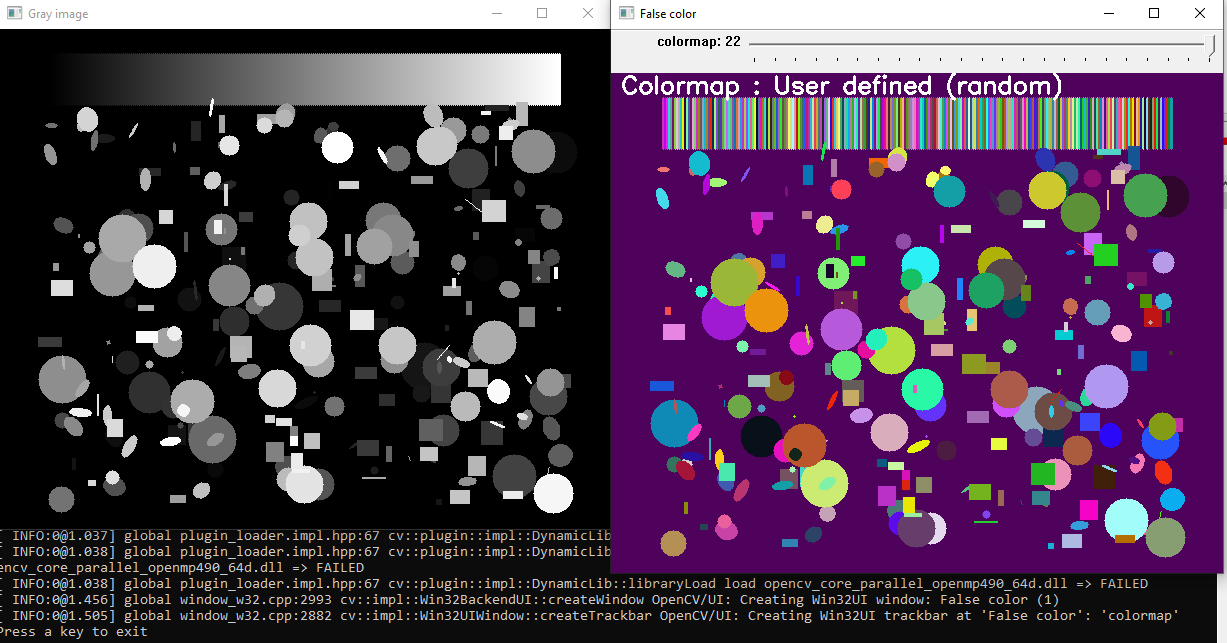


Рисунок 18 – Цветовая карта – User defined

## **create\_mask.cpp**

|  |  |  |
| --- | --- | --- |
| **Описание идиомы** | **Номер строк кода** | **Комментарий** |
| [Включение стандартных библиотек](https://hyperpolyglot.org/cpp#implicit-prologue-note) | 12-15 |  |
| [Директивы пространства имён](https://hyperpolyglot.org/cpp#import-namespace) | 17-18 | На 6 строке использовано пространство имён, чтобы избежать постоянного написания cv перед методами библиотеки |
| [Объявление переменной типа](https://hyperpolyglot.org/cpp#local-var-note) [Mat](https://docs.opencv.org/4.x/d3/d63/classcv_1_1Mat.html) | 20 | Создает переменную типа mat. Она может хранить в себе векторы или матрицы, а также изображения |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа point](https://docs.opencv.org/4.x/db/d4e/classcv_1_1Point__.html) | 22 | Класс шаблона для 2D-точек, заданных их координатами x и y. |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа vector](https://hyperpolyglot.org/cpp#decl-resizable-array-note) | 23 |  |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа int](https://hyperpolyglot.org/cpp#int-type-note) | 24-26 |  |
| [Объявление функции](https://hyperpolyglot.org/cpp#decl-func) | 28, 30, 94 | 28,30 – обработчик нажатий мыши  94 – вход в приложение |
| [Условный оператор if](https://hyperpolyglot.org/cpp#if-note) | 33, 35, 37, 45, 52, 58, 63, 71, 84, 106 |  |
| [circle](https://docs.opencv.org/4.x/d9/db7/group__datasets__gr.html#gga610754124ced68d1f05760b5948fbb76a6f0d8b2d9e3e947b2a5c1eff9e81ee95) | 40, | Рисует круг |
| [clone](https://docs.opencv.org/4.x/d3/d63/classcv_1_1Mat.html#a03d2a2570d06dcae378f788725789aa4) | 38, 61 | Создает полную копию массива и базовых данных. |
| [push\_back](https://hyperpolyglot.org/cpp#resizable-array-back-note) | 41 |  |
| x[++](https://hyperpolyglot.org/cpp#indexed-array-iteration-note) | 42 |  |
| [line](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga7078a9fae8c7e7d13d24dac2520ae4a2) | 46 | Рисует линию от одной точки к другой |
| [imshow](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#ga453d42fe4cb60e5723281a89973ee563) | 48, 54, 68, 79, 80, 81, 90, 114 | Отображает изображение в указанном окне. |
| [polylines](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#gaa3c25f9fb764b6bef791bf034f6e26f5) | 65 | Рисует несколько многоугольных кривых. |
| [fillPoly](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga311160e71d37e3b795324d097cb3a7dc) | 77 | Заполняет область, ограниченную одним или несколькими полигонами. |
| [bitwise\_and](https://docs.opencv.org/4.x/d2/de8/group__core__array.html#ga60b4d04b251ba5eb1392c34425497e14) | 78 | вычисляет поразрядное объединение двух массивов (dst = src1 и src2). Вычисляет побитовое объединение двух элементов или массива и скаляра. |
| [CommandLineParser](https://docs.opencv.org/4.x/d0/d2e/classcv_1_1CommandLineParser.html) | 96 | Предназначен для анализа командной строки. |
| [imread](https://docs.opencv.org/4.x/d4/da8/group__imgcodecs.html#gab32ee19e22660912565f8140d0f675a8) | 104 | Загружает изображение из файла. |
| [printf](https://hyperpolyglot.org/cpp#printf-note) | 108 |  |
| [namedWindow](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#ga5afdf8410934fd099df85c75b2e0888b) | 112 | Создает окно.  Функция NameWindow создает окно, которое можно использовать в качестве заполнителя для изображений и трекбаров. Созданные окна называются по именам.  Если окно с таким именем уже существует, функция ничего не делает. |
| [setMouseCallback](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#ga89e7806b0a616f6f1d502bd8c183ad3e) | 113 | Устанавливает обработчик мыши для указанного окна. |
| [waitKey](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#ga5628525ad33f52eab17feebcfba38bd7) | 115 | Ожидает нажатия клавиши. |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа string](https://hyperpolyglot.org/cpp#str-type) | 102 |  |
| [Оператор возврата значения из функции](https://hyperpolyglot.org/cpp#retval) | 109, 117 |  |

1. /\*

2. \* create\_mask.cpp

3. \*

4. \* Author:

5. \* Siddharth Kherada <siddharthkherada27[at]gmail[dot]com>

6. \*

7. \* This tutorial demonstrates how to make mask image (black and white).

8. \* The program takes as input a source image and outputs its corresponding

9. \* mask image.

10. \*/

11.

12. #include "opencv2/imgproc.hpp"

13. #include "opencv2/imgcodecs.hpp"

14. #include "opencv2/highgui.hpp"

15. #include <iostream>

16.

17. using namespace std;

18. using namespace cv;

19.

20. Mat src, img1, mask, final;

21.

22. Point point;

23. vector<Point> pts;

24. int drag = 0;

25. int var = 0;

26. int flag = 0;

27.

28. void mouseHandler(int, int, int, int, void\*);

29.

30. void mouseHandler(int event, int x, int y, int, void\*)

31. {

32.

33. if (event == EVENT\_LBUTTONDOWN && !drag)

34. {

35. if (flag == 0)

36. {

37. if (var == 0)

38. img1 = src.clone();

39. point = Point(x, y);

40. circle(img1, point, 2, Scalar(0, 0, 255), -1, 8, 0);

41. pts.push\_back(point);

42. var++;

43. drag = 1;

44.

45. if (var > 1)

46. line(img1, pts[var - 2], point, Scalar(0, 0, 255), 2, 8, 0);

47.

48. imshow("Source", img1);

49. }

50. }

51.

52. if (event == EVENT\_LBUTTONUP && drag)

53. {

54. imshow("Source", img1);

55. drag = 0;

56. }

57.

58. if (event == EVENT\_RBUTTONDOWN)

59. {

60. flag = 1;

61. img1 = src.clone();

62.

63. if (var != 0)

64. {

65. polylines(img1, pts, 1, Scalar(0, 0, 0), 2, 8, 0);

66. }

67.

68. imshow("Source", img1);

69. }

70.

71. if (event == EVENT\_RBUTTONUP)

72. {

73. flag = var;

74. final = Mat::zeros(src.size(), CV\_8UC3);

75. mask = Mat::zeros(src.size(), CV\_8UC1);

76.

77. fillPoly(mask, pts, Scalar(255, 255, 255), 8, 0);

78. bitwise\_and(src, src, final, mask);

79. imshow("Mask", mask);

80. imshow("Result", final);

81. imshow("Source", img1);

82. }

83.

84. if (event == EVENT\_MBUTTONDOWN)

85. {

86. pts.clear();

87. var = 0;

88. drag = 0;

89. flag = 0;

90. imshow("Source", src);

91. }

92. }

93.

94. int main(int argc, char\*\* argv)

95. {

96. CommandLineParser parser(argc, argv, "{@input | lena.jpg | input image}");

97. parser.about("This program demonstrates using mouse events\n");

98. parser.printMessage();

99. cout << "\n\tleft mouse button - set a point to create mask shape\n"

100. "\tright mouse button - create mask from points\n"

101. "\tmiddle mouse button - reset\n";

102. String input\_image = parser.get<String>("@input");

103.

104. src = imread(samples::findFile(input\_image));

105.

106. if (src.empty())

107. {

108. printf("Error opening image: %s\n", input\_image.c\_str());

109. return 0;

110. }

111.

112. namedWindow("Source", WINDOW\_AUTOSIZE);

113. setMouseCallback("Source", mouseHandler, NULL);

114. imshow("Source", src);

115. waitKey(0);

116.

117. return 0;

118. }

Пример работы приложения(рис. 19-21):



Рисунок 19 – Запуск программы и выделение объекта с помощью левой клавиши мыши.



Рисунок 20 – Создание маски из выделенных участков изображения с помощью нажатия правой кнопки мыши.



Рисунок 21 – Итоговый результат

## **Drawing\_2.cpp**

|  |  |  |
| --- | --- | --- |
| **Описание идиомы** | **Номер строк кода** | **Комментарий** |
| [Включение стандартных библиотек](https://hyperpolyglot.org/cpp#implicit-prologue-note) | 2-6 |  |
| [Директивы пространства имён](https://hyperpolyglot.org/cpp#import-namespace) | 8 | На 6 строке использовано пространство имён, чтобы избежать постоянного написания cv перед методами библиотеки |
| [объявление функции типа int](https://hyperpolyglot.org/cpp#global-var-note) | 21-28, 31, 79, 101, 126, 156, 190, 224, 247, 270, |  |
| [Глобальное объявление переменной типа int](https://hyperpolyglot.org/cpp#global-var-note) | 10-11, 13-18 |  |
| [объявление функции](https://hyperpolyglot.org/cpp#global-var-note) [типа Scalar](https://docs.opencv.org/4.x/d1/da0/classcv_1_1Scalar__.html) | 20, 72 | Класс шаблона для 4-элементного вектора, полученного из [**Vec**](https://docs.opencv.org/4.x/d6/dcf/classcv_1_1Vec.html) . |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа int](https://hyperpolyglot.org/cpp#int-type-note) | 33, 74, 104, 105, 128, 158, 177, 192, 211, 226, 249, 274, |  |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа char](https://hyperpolyglot.org/cpp#str-type-note) | 35 |  |
| [RNG](https://docs.opencv.org/4.x/d2/d44/classcv_1_1Rect__.html) | 37, | Генератор 2D-прямоугольников |
| [Объявление переменной типа](https://hyperpolyglot.org/cpp#local-var-note) [Mat](https://docs.opencv.org/4.x/d3/d63/classcv_1_1Mat.html) | 39, 276 | Создает переменную типа mat. Она может хранить в себе векторы или матрицы, а также изображения |
| [imshow](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#ga453d42fe4cb60e5723281a89973ee563) | 40, 91, 116, 145, 181, 215, 237, 260, 284 | Отображает изображение в указанном окне. |
| [waitKey](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html#ga5628525ad33f52eab17feebcfba38bd7) | 41, 67, | Ожидает нажатия клавиши. |
| [Условный оператор if](https://hyperpolyglot.org/cpp#if-note) | 44, 47, 50, 53, 56, 59, 62, 65, 92, 117, 147, 182, 216, 238, 261, 285 |  |
| [Оператор возврата значения из функции](https://hyperpolyglot.org/cpp#retval) | 44, 47, 50, 53, 56, 59, 62, 65, 68, 75, 94, 98, 119, 123, 149, 153, 184, 187, 218, 221, 240, 244, 263, 267, 287, 291 |  |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа point](https://docs.opencv.org/4.x/db/d4e/classcv_1_1Point__.html) | 81, 103, 132, 162, 176, 196, 210, 230, 253, 273 | Класс шаблона для 2D-точек, заданных их координатами x и y. |
| [Цикл for](https://hyperpolyglot.org/cpp#for-note) | 83, 107, 130, 160, 194, 228, 251, 278 |  |
| [uniform](https://docs.opencv.org/4.x/d1/dd6/classcv_1_1RNG.html#acde197860cea91e5aa896be8719457ae) | 85-88, 109-112, 163-174 … | возвращает равномерно распределенное целое случайное число из диапазона [a,b) |
| [line](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga7078a9fae8c7e7d13d24dac2520ae4a2) | 90 | Рисует линию от одной точки к другой |
| [rectangle](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga07d2f74cadcf8e305e810ce8eed13bc9) | 114 | Рисует прямоугольник |
| [Объявление переменной](https://hyperpolyglot.org/cpp#local-var-note) [типа Size](https://docs.opencv.org/4.x/d6/d50/classcv_1_1Size__.html) | 132, 258 | Класс шаблона для указания размера изображения или прямоугольника. |
| [ellipse](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga28b2267d35786f5f890ca167236cbc69) | 138 | Рисует эллипс |
| [polylines](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#gaa3c25f9fb764b6bef791bf034f6e26f5) | 173 | Рисует несколько многоугольных кривых. |

1. #include <opencv2/core.hpp>

2. #include <opencv2/imgproc.hpp>

3. #include <opencv2/highgui.hpp>

4. #include <iostream>

5. #include <stdio.h>

6.

7. using namespace cv;

8.

9. const int NUMBER = 100;

10. const int DELAY = 5;

11.

12. const int window\_width = 900;

13. const int window\_height = 600;

14. int x\_1 = -window\_width/2;

15. int x\_2 = window\_width\*3/2;

16. int y\_1 = -window\_width/2;

17. int y\_2 = window\_width\*3/2;

18.

19. static Scalar randomColor( RNG& rng );

20. int Drawing\_Random\_Lines( Mat image, char\* window\_name, RNG rng );

21. int Drawing\_Random\_Rectangles( Mat image, char\* window\_name, RNG rng );

22. int Drawing\_Random\_Ellipses( Mat image, char\* window\_name, RNG rng );

23. int Drawing\_Random\_Polylines( Mat image, char\* window\_name, RNG rng );

24. int Drawing\_Random\_Filled\_Polygons( Mat image, char\* window\_name, RNG rng );

25. int Drawing\_Random\_Circles( Mat image, char\* window\_name, RNG rng );

26. int Displaying\_Random\_Text( Mat image, char\* window\_name, RNG rng );

27. int Displaying\_Big\_End( Mat image, char\* window\_name, RNG rng );

28.

29.

30. int main( void )

31. {

32. int c;

33.

34. char window\_name[] = "Drawing\_2 Tutorial";

35.

36. RNG rng( 0xFFFFFFFF );

37.

38. Mat image = Mat::zeros( window\_height, window\_width, CV\_8UC3 );

39. imshow( window\_name, image );

40. waitKey( DELAY );

41.

42. c = Drawing\_Random\_Lines(image, window\_name, rng);

43. if( c != 0 ) return 0;

44.

45. c = Drawing\_Random\_Rectangles(image, window\_name, rng);

46. if( c != 0 ) return 0;

47.

48. c = Drawing\_Random\_Ellipses( image, window\_name, rng );

49. if( c != 0 ) return 0;

50.

51. c = Drawing\_Random\_Polylines( image, window\_name, rng );

52. if( c != 0 ) return 0;

53.

54. c = Drawing\_Random\_Filled\_Polygons( image, window\_name, rng );

55. if( c != 0 ) return 0;

56.

57. c = Drawing\_Random\_Circles( image, window\_name, rng );

58. if( c != 0 ) return 0;

59.

60. c = Displaying\_Random\_Text( image, window\_name, rng );

61. if( c != 0 ) return 0;

62.

63. c = Displaying\_Big\_End( image, window\_name, rng );

64. if( c != 0 ) return 0;

65.

66. waitKey(0);

67. return 0;

68. }

69.

70.

71. static Scalar randomColor( RNG& rng )

72. {

73. int icolor = (unsigned) rng;

74. return Scalar( icolor&255, (icolor>>8)&255, (icolor>>16)&255 );

75. }

76.

77.

78. int Drawing\_Random\_Lines( Mat image, char\* window\_name, RNG rng )

79. {

80. Point pt1, pt2;

81.

82. for( int i = 0; i < NUMBER; i++ )

83. {

84. pt1.x = rng.uniform( x\_1, x\_2 );

85. pt1.y = rng.uniform( y\_1, y\_2 );

86. pt2.x = rng.uniform( x\_1, x\_2 );

87. pt2.y = rng.uniform( y\_1, y\_2 );

88.

89. line( image, pt1, pt2, randomColor(rng), rng.uniform(1, 10), 8 );

90. imshow( window\_name, image );

91. if( waitKey( DELAY ) >= 0 )

92. { return -1; }

93. }

94.

95. return 0;

96. }

97.

98. int Drawing\_Random\_Rectangles( Mat image, char\* window\_name, RNG rng )

99. {

100. Point pt1, pt2;

101. int lineType = 8;

102. int thickness = rng.uniform( -3, 10 );

103.

104. for( int i = 0; i < NUMBER; i++ )

105. {

106. pt1.x = rng.uniform( x\_1, x\_2 );

107. pt1.y = rng.uniform( y\_1, y\_2 );

108. pt2.x = rng.uniform( x\_1, x\_2 );

109. pt2.y = rng.uniform( y\_1, y\_2 );

110.

111. rectangle( image, pt1, pt2, randomColor(rng), MAX( thickness, -1 ), lineType );

112.

113. imshow( window\_name, image );

114. if( waitKey( DELAY ) >= 0 )

115. { return -1; }

116. }

117.

118. return 0;

119. }

120.

121. int Drawing\_Random\_Ellipses( Mat image, char\* window\_name, RNG rng )

122. {

123. int lineType = 8;

124.

125. for ( int i = 0; i < NUMBER; i++ )

126. {

127. Point center;

128. center.x = rng.uniform(x\_1, x\_2);

129. center.y = rng.uniform(y\_1, y\_2);

130.

131. Size axes;

132. axes.width = rng.uniform(0, 200);

133. axes.height = rng.uniform(0, 200);

134.

135. double angle = rng.uniform(0, 180);

136.

137. ellipse( image, center, axes, angle, angle - 100, angle + 200,

138. randomColor(rng), rng.uniform(-1,9), lineType );

139.

140. imshow( window\_name, image );

141.

142. if( waitKey(DELAY) >= 0 )

143. { return -1; }

144. }

145.

146. return 0;

147. }

148.

149. int Drawing\_Random\_Polylines( Mat image, char\* window\_name, RNG rng )

150. {

151. int lineType = 8;

152.

153. for( int i = 0; i< NUMBER; i++ )

154. {

155. Point pt[2][3];

156. pt[0][0].x = rng.uniform(x\_1, x\_2);

157. pt[0][0].y = rng.uniform(y\_1, y\_2);

158. pt[0][1].x = rng.uniform(x\_1, x\_2);

159. pt[0][1].y = rng.uniform(y\_1, y\_2);

160. pt[0][2].x = rng.uniform(x\_1, x\_2);

161. pt[0][2].y = rng.uniform(y\_1, y\_2);

162. pt[1][0].x = rng.uniform(x\_1, x\_2);

163. pt[1][0].y = rng.uniform(y\_1, y\_2);

164. pt[1][1].x = rng.uniform(x\_1, x\_2);

165. pt[1][1].y = rng.uniform(y\_1, y\_2);

166. pt[1][2].x = rng.uniform(x\_1, x\_2);

167. pt[1][2].y = rng.uniform(y\_1, y\_2);

168.

169. const Point\* ppt[2] = {pt[0], pt[1]};

170. int npt[] = {3, 3};

171.

172. polylines(image, ppt, npt, 2, true, randomColor(rng), rng.uniform(1,10), lineType);

173.

174. imshow( window\_name, image );

175. if( waitKey(DELAY) >= 0 )

176. { return -1; }

177. }

178. return 0;

179. }

180.

181. int Drawing\_Random\_Filled\_Polygons( Mat image, char\* window\_name, RNG rng )

182. {

183. int lineType = 8;

184.

185. for ( int i = 0; i < NUMBER; i++ )

186. {

187. Point pt[2][3];

188. pt[0][0].x = rng.uniform(x\_1, x\_2);

189. pt[0][0].y = rng.uniform(y\_1, y\_2);

190. pt[0][1].x = rng.uniform(x\_1, x\_2);

191. pt[0][1].y = rng.uniform(y\_1, y\_2);

192. pt[0][2].x = rng.uniform(x\_1, x\_2);

193. pt[0][2].y = rng.uniform(y\_1, y\_2);

194. pt[1][0].x = rng.uniform(x\_1, x\_2);

195. pt[1][0].y = rng.uniform(y\_1, y\_2);

196. pt[1][1].x = rng.uniform(x\_1, x\_2);

197. pt[1][1].y = rng.uniform(y\_1, y\_2);

198. pt[1][2].x = rng.uniform(x\_1, x\_2);

199. pt[1][2].y = rng.uniform(y\_1, y\_2);

200.

201. const Point\* ppt[2] = {pt[0], pt[1]};

202. int npt[] = {3, 3};

203.

204. fillPoly( image, ppt, npt, 2, randomColor(rng), lineType );

205.

206. imshow( window\_name, image );

207. if( waitKey(DELAY) >= 0 )

208. { return -1; }

209. }

210. return 0;

211. }

212.

213. int Drawing\_Random\_Circles( Mat image, char\* window\_name, RNG rng )

214. {

215. int lineType = 8;

216.

217. for (int i = 0; i < NUMBER; i++)

218. {

219. Point center;

220. center.x = rng.uniform(x\_1, x\_2);

221. center.y = rng.uniform(y\_1, y\_2);

222.

223. circle( image, center, rng.uniform(0, 300), randomColor(rng),

224. rng.uniform(-1, 9), lineType );

225.

226. imshow( window\_name, image );

227. if( waitKey(DELAY) >= 0 )

228. { return -1; }

229. }

230.

231. return 0;

232. }

233.

234. int Displaying\_Random\_Text( Mat image, char\* window\_name, RNG rng )

235. {

236. int lineType = 8;

237.

238. for ( int i = 1; i < NUMBER; i++ )

239. {

240. Point org;

241. org.x = rng.uniform(x\_1, x\_2);

242. org.y = rng.uniform(y\_1, y\_2);

243.

244. putText( image, "Testing text rendering", org, rng.uniform(0,8),

245. rng.uniform(0,100)\*0.05+0.1, randomColor(rng), rng.uniform(1, 10), lineType);

246.

247. imshow( window\_name, image );

248. if( waitKey(DELAY) >= 0 )

249. { return -1; }

250. }

251.

252. return 0;

253. }

254.

255. int Displaying\_Big\_End( Mat image, char\* window\_name, RNG )

256. {

257. Size textsize = getTextSize("OpenCV forever!", FONT\_HERSHEY\_COMPLEX, 3, 5, 0);

258. Point org((window\_width - textsize.width)/2, (window\_height - textsize.height)/2);

259. int lineType = 8;

260.

261. Mat image2;

262.

263. for( int i = 0; i < 255; i += 2 )

264. {

265. image2 = image - Scalar::all(i);

266. putText( image2, "OpenCV forever!", org, FONT\_HERSHEY\_COMPLEX, 3,

267. Scalar(i, i, 255), 5, lineType );

268.

269. imshow( window\_name, image2 );

270. if( waitKey(DELAY) >= 0 )

271. { return -1; }

272. }

273.

274. return 0;

275. }

Пример работы приложения(рис. 22-23):

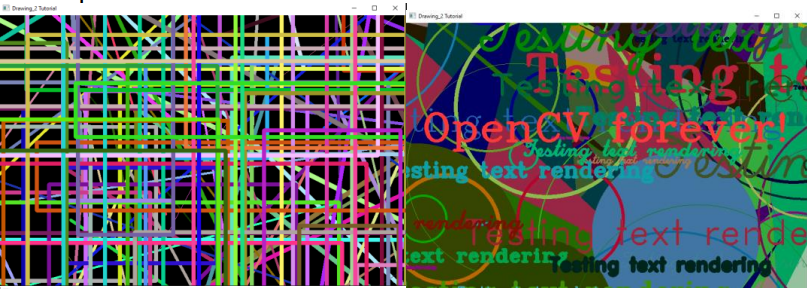
Рисунок 22 – Анимация из случайных линий и фигур



Рисунок 23 – Конец анимации

# Задание 3.

1. [**Избегать использования strdup()**](https://www.olivierlanglois.net/idioms_for_using_cpp_in_c_programs.html)

Эта функция не является стандартной для C и C++, поэтому её поведение может быть неожиданным.

Требуемое возвращаемое значение – указатель на копию строки. Если выделение памяти закончилось неудачей, то создания копии не происходит и функция возвращает NULL.

Чтобы избежать такой проблемы можно переопределить strdup() в пространстве имён.

namespace util

{

// Variant #1

char \*strdup(const char \*s, int size)

{

++size;

char \*res = static\_cast<char \*>(malloc(size));

if(res)

{

// Use memcpy() as it is faster than strcpy()

// since we already have the strlen.

memcpy(res,s,size);

}

return res;

}

// Variant #2

char \*strdup(const char \*s)

{

return strdup(s,strlen(s));

}

}