

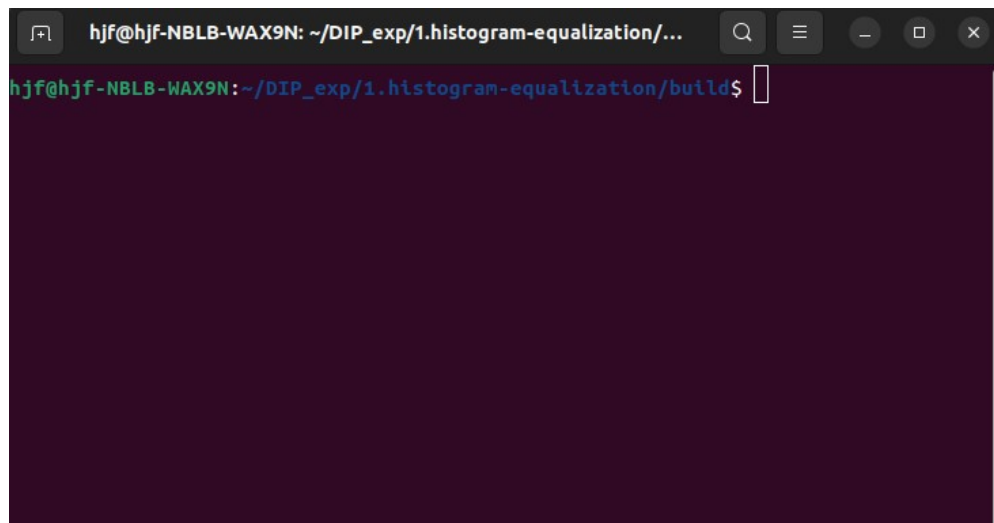
南工骁鹰视觉组-C++培训

- 图像分通道显示
- 笔试-寻找矩形

目录

- 1. 入门程序 - Hello World
- 2. 环境配置的简单说明
- 3. 链接与编译的基础概念
- 4. C++基础语法
- 5. 类与模板
- 6. 一些新特性
- 7. 小练习

- Terminal
- 是专门为程序员设计的，通过输入命令来操作电脑的一种方式



Hello World

- `#include <iostream>`
- `using namespace std;`
- `int main(void)`
- `{`
 - `cout << "Hello World!" << endl;`
 - `return 0;`
- `}`

C++程序由一个或多个被称为函数的模块组成。程序从 `main()` 函数（全部小写）开始执行，因此该函数必不可少。函数由函数头和函数体组成。函数头指出函数的返回值（如果有的话）的类型和函数期望通过参数传递给它的信息的类型。函数体由一系列位于花括号（`{}`）中的 C++ 语句组成。

有多种类型的 C++ 语句，包括下述 6 种。

- 声明语句：定义函数中使用的变量的名称和类型。
- 赋值语句：使用赋值运算符（`=`）给变量赋值。
- 消息语句：将消息发送给对象，激发某种行动。
- 函数调用：执行函数。被调用的函数执行完毕后，程序返回到函数调用语句后面的语句。
- 函数原型：声明函数的返回类型、函数接受的参数数量和类型。
- 返回语句：将一个值从被调用的函数那里返回到调用函数中。

类是用户定义的数据类型规范，它详细描述了如何表示信息以及可对数据执行的操作。对象是根据类规范创建的实体，就像简单变量是根据数据类型描述创建的实体一样。

C++ 提供了两个用于处理输入和输出的预定义对象（`cin` 和 `cout`），它们是 `istream` 和 `ostream` 类的实例，这两个类是在 `iostream` 文件中定义的。为 `ostream` 类定义的插入运算符（`<<`）使得将数据插入到输出流成为可能；为 `istream` 类定义的抽取运算符（`>>`）能够从输入流中抽取信息。`cin` 和 `cout` 都是智能对象，能够根据程序上下文自动将信息从一种形式转换为另一种形式。

C++ 可以使用大量的 C 库函数。要使用库函数，应当包含提供该函数原型的头文件。

环境配置

- 文本编辑器 : vi vim gedit nano vscode
- 编译器: gcc g++ clang
- 调试器: gdb
-
- IDE: 集成开发环境-Visual Studio、CLion

vim的基本使用

- 命令模式
- 视图模式 v-进入视图模式
- 插入模式
- u-撤销 | x-删除当前字符 | ndd-剪切n行 | ngg-跳转到n行 | p-粘贴
- :wq - 保存并退出 :set nu 显示行号
- :q! - 强制退出

```
20      for (int j = 0; j < N; j++)
21      {
22          arr[i][j] = (cin.get())
23      }
:wq
```


编译与链接

- 预处理 `#define ...` | `gcc -E`
- 编译 | `gcc -S`
- 汇编 | `gcc -c`
- 链接

```
-E          Preprocess only; do not compile, assemble or link.
-S          Compile only; do not assemble or link.
-c          Compile and assemble, but do not link.
-o <file>   Place the output into <file>.
```

基本的类型与运算

- char int float double
- 算术运算符 + - * /
- 关系运算符 != == >= <=
- 逻辑运算符 && || !
- 位运算符 ^ & | -

控制语句

- if
- switch
- while
- for(int i=0; i < number; i++)
-
- break
- continue
- return

函数的使用

- 形参与返回值
- `const` 修饰
- `static` 修饰
- `lambda`表达式

指针与引用

- * 变量-地址
- & 取地址运算符 * 指针运算符
- -----
- & 必须初始化
- 为变量取别名
- const type & 函数传参

编译工具

- gcc
- g++
- clang
-
- Makefile
- CMake

- build目录-
- 外部编译与内部编译

```
cmake_minimum_required(VERSION 3.10)
project(Histogram)

find_package(OpenCV REQUIRED)
include_directories(${OpenCV_INCLUDE_DIRS})
message(STATUS ${OpenCV_INCLUDE_DIRS})

add_executable(histogram src/histogram.cpp)
target_link_libraries(histogram ${OpenCV_LIBS})
```

- `cmake --version`
- 终端执行此命令，查看cmake版本
-
- `cmake_minimum_required()`
- 要求cmake的最低版本
-

- `project()`
- 指定项目名称
-
- `aux_source_directory(. SRC_LIST)`
- 将指定目录的文件存放到变量中

- `add_compile_options(-std=c++11)`
- 添加编译选项
-
- `include_directories(./include)`
- 添加头文件搜索路径

- `add_executable(main main.cpp)`
- 添加源文件以生成可执行文件
-
- `add_subdirectory(./src)`
- 添加子目录，需要子目录也有CMakeLists.txt文件以进行编译
-
- `set()` 设置变量

- `add_library(mylib ${source})`
- 根据源文件生成库
-
- `find_library(TEST_LIB testfunc HINTS ./lib)`
- 在指定目录下寻找库，并将路径放入变量中
-
- `target_link_libraries(main my_lib)` 链接对应文件

类的使用

- Public
- Private
-
- 抽象、封装、继承与多态
- 构造与析构
- 运算符重载
- this指针

- 默认的六个成员函数
- 构造函数、析构函数
- 拷贝构造函数*（按值传入、返回，赋值初始化）
- =赋值运算符重载、&取地址符重载、const修饰的取地址符重载

- 纯虚函数 没有实现的虚函数
- 抽象类 包含虚函数的类
- 派生类 必须实现基类的所有纯虚函数
-
- Override 检查是否重写基类的虚函数
- Final 修饰类、虚函数-不可继承、重载

模板

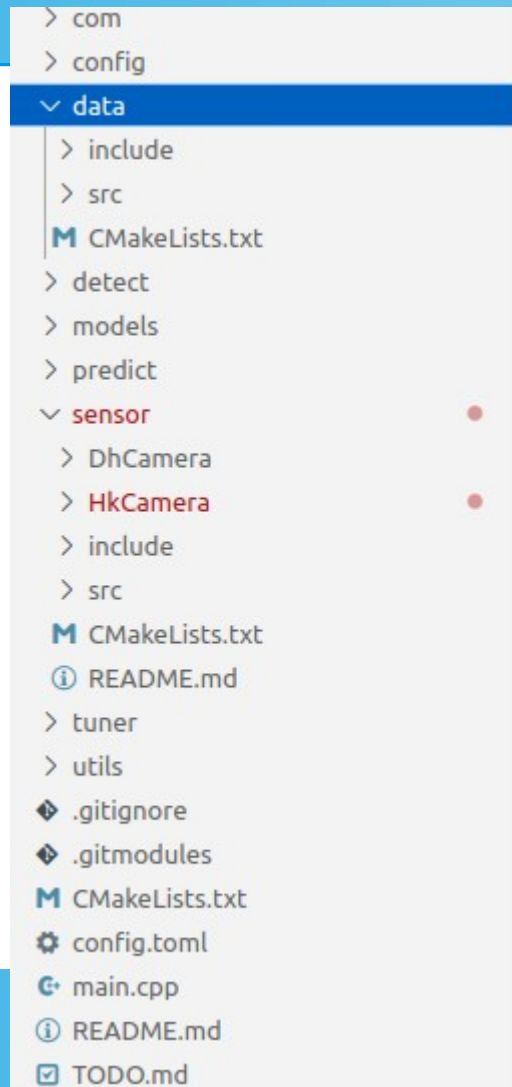
- `template <typename T>`
-
- 函数模板
- 类模板

案例

- 复数类的实现
- 直方图均衡化

工程规范

- 工程结构
-
- 代码规范
-



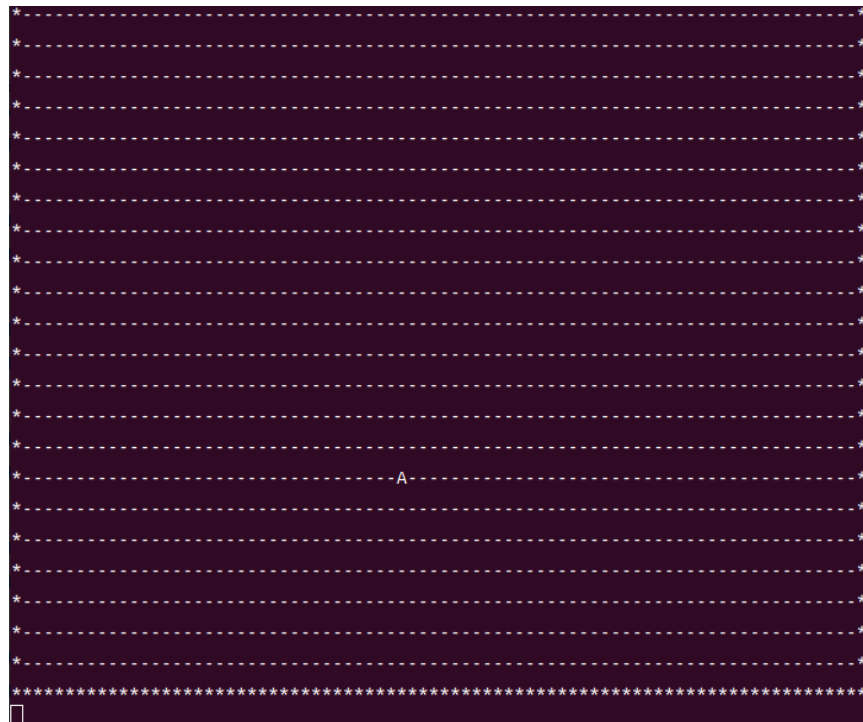
声明与定义

- `int a;`
- `int b=2;`
- `extern int c;`
-
- `class myclass;`
- `int func(void);`

一些新特性

- auto
- for循环
- decltype 泛型编程中结合auto，用于追踪函数的返回值类型
-

小练习-keybot



练习

- 应用面向对象思想，编写机器人类、地图绘制类与地图进行交互
- 可通过键盘控制机器人的移动
-
- 进阶：使用虚类，实现不同属性的机器人
- 为地图绘制类提供更多接口，使其可以更方便地编辑地图
- 了解寻路算法，尝试为机器人实现寻路功能

练习提交

- 将工程打包，命名格式：姓名-学号
- 要求使用cmake管理编译，工程结构明确，代码遵循基本规范
-
- 于10.7日前发送至邮箱1360915472@qq.com