برنامه‌نویسی جریان داده یا Data Flow Programming یک پارادایم برنامه‌نویسی است که در آن برنامه‌ها به صورت گرافیکی از توابع و اشیاء ساخته شده‌اند که در آن داده از یک نقطه به نقطه دیگر در طی زمان حرکت می‌کند. این شیوه برنامه‌نویسی بر پایه انتقال داده‌ها از یک مرحله به مرحله دیگر تاکید دارد و اجزاء برنامه به صورت موازی اجرا می‌شوند.

یک مثال ساده از برنامه‌نویسی جریان داده در C++ با استفاده از کتابخانه OpenCV در زیر آمده است، یک تصویر از دوربین گرفته می‌شود و سپس با فیلترسازی آن، تصویر خروجی تولید می‌شود.

#include <opencv2/opencv.hpp>

int main() {

    // خواندن تصویر از دوربین

    cv::VideoCapture capture(0);

    if (!capture.isOpened()) {

        std::cerr << "Error: Could not open camera." << std::endl;

        return -1;

    }

    cv::Mat inputImage, outputImage;

    while (true) {

        // خواندن فریم از دوربین

        capture >> inputImage;

        // اعمال یک فیلتر (برای مثال: فیلتر سیاه و سفید)

        cv::cvtColor(inputImage, outputImage, cv::COLOR\_BGR2GRAY);

        // نمایش تصویر خروجی

        cv::imshow("Output Image", outputImage);

        // انتظار برای کلیدهای فشرده شده

        if (cv::waitKey(30) == 27) {

            break;

        }

    }

    return 0;

}

```

کاربردهای این پارادایم به شرح زیر است:

* پردازش تصویر و گرافیک:

برنامه‌نویسی جریان داده در حوزه پردازش تصویر و گرافیک بسیار موثر است، زیرا عملیات مختلف مانند فیلترسازی، تحلیل تصویر و تصویرسازی را به صورت موازی مدیریت می‌کند.

* پردازش سیگنال:

در حوزه سیگنال‌ها، مانند پردازش صوت یا سیگنال‌های الکترونیکی، برنامه‌نویسی جریان داده به خوبی قابل استفاده است.

* شبیه‌سازی و مدل‌سازی:

در شبیه‌سازی و مدل‌سازی سیستم‌های پیچیده، برنامه‌نویسی جریان داده به تدریج محبوب‌تر شده است.

مزایای آن به شرح زیر است:

* پویایی و انعطاف‌پذیری:

این شیوه برنامه‌نویسی به برنامه‌نویس این امکان را می‌دهد که به راحتی اجزاء مختلف را اضافه، حذف یا تغییر دهد

* موازی‌سازی ساده‌تر: با تأکید بر جریان داده، برنامه‌نویسی موازی‌سازی ساده‌تر و شفاف‌تر است.

علتوه بر این مزایا برخی معایب نیز دارد:

* پنهان کننده ساختار:

گاهی اوقات یک گراف جریان داده می‌تواند پیچیده و دشوار برای فهم باشد، به ویژه برای برنامه‌های کوچک

* بهینه‌سازی دشوارتر

بهینه‌سازی برنامه‌های بزرگ ممکن است چالشی باشد زیرا گاهی اوقات نیاز به بهینه‌سازی دقیق در جریان داده وابسته به ویژگی‌های خاص برنامه دارد.

برنامه‌نویسی جریان داده بیشتر در حوزه‌های خاص مانند پردازش تصویر و صوت، شبیه‌سازی و مدل‌سازی، و برنامه‌های سیگنال مورد استفاده قرار می‌گیرد که نیاز به پردازش همزمان داده‌های بزرگ دارند.