



**زیربرنامه:**

Surface.cpp

|  |  |  |
| --- | --- | --- |
| **توسعه دهندگان** | مرتضی نامور |  |
| **تهیه کنندگان مستند** | مرتضی نامور | |
| **تاییدکنندگان** | مرتضی نامور | |
| **تاریخ تنظیم سند** | 22/02/1394 | |
| **شناسه سند** | **MC2F003F1** | |
| **زبان برنامه‌نویسی** | **Fortran 90** | |

1. وظایف

هدف از تعریف این کلاس نگه داری اضلاع و المان های یک لایه در کنار هم است به گونه ای که بتوان یک لایه دیگر در کنار آن اضافه کرد یا اینکه تمام آن لایه را از مجموعه شبکه هندسی حذف کرد. همانطور که می دانیم یک لایه متشکل از تعدادی ضلع و المان های همسایه آن اضلاع است. در ارتباط با چگونگی تشکیل یک لایه در کلاس ZONE بحث خواهیم کرد. هم اکنون فقط به تعریف مشخصات و ویژگی های یک کلاس surface می پردازیم.

1. توضیحات و تئوری­ها

بدون تئوری

1. بخش­های زیربرنامه
2. تعریف متد order

این متد خروجی ندارد و از نوع void تعریف شده است. هدف از تعریف این متد مرتب کردن اضلاع و المان های یک لایه پشت سر هم است. برای بهتر متوجه شدن عملکرد این متد باید به نحوه استخراج یک لایه که در متد extend از کلاس ZONE مراجعه شود. به طور اختصار در متد extend برای استخراج لایه ابتدا باید یک شماره ضلعی وارد شود. متد این ضلع را دریافت کرده و اضلاع متصل و هم راستای ضلع ورودی را استخراج می کند. از آنجایی که ضلع ورودی می تواند در وسط یک لایه قرار گرفته باشد آن گاه این متد به طور مثال اول اضلاع سمت راست این ضلع را استخراج می کند و پس از اتمام آن ها به سراغ اضلاع سمت چپ می رود. در اینجا بخاطر نحوه عملکرد متد خود ضلع ورودی دو مرتبه در لایه تکرار می شود. یک بار به عنوان اولین ضلع ورودی که در ابتدای vector E قرار می گیرد و بار دوم پس از اتمام اضلاع یک سمت و برای شروع اضلاع سمت دیگر در میانه های vector E قرار می گیرد. ما از این اتفاق برای مرتب کردن اضلاع یک لایه پشت سر هم استفاده می کنیم. برای روشن تر شدن ذهن فرض کنید که یک لایه با 10 ضلع به صورت زیر وجود دارد:

9

8

7

6

5

4

3

2

1

10

1. نمایش یک لایه به همراه اضلاع و المان های آن

در شکل فوق خطوط قرمز رنگ اضلاع لایه مورد نظر هستند و المان های دو طرف نیز المان های همسایه می باشند. حال فرض کنید اگر ضلع 4 به عنوان ورودی به متد extend برای استخراج لایه داده شود خروجی آن متد یک vector به صورت زیر خواهد بود:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 10 | 9 | 8 | 7 | 6 | 5 | 4 |

1. خروجی متد extend

توجه شود که در آرایه فوق عدد 4 که به عنوان ضلع ورودی داده شده است دو مرتبه تکرار می شود.

ما با استفاده از متد order قصد داریم که خروجی فوق را به صورت زیر در بیاوریم:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

1. خروجی متد order

در این vector جدید هم درایه ها به صورت مرتب پشت سر هم قرار گرفته اند و هم اینکه اندازه vector یکی کمتر از حالت قبل است زیرا خود ضلع 4 که دو مرتبه تکرار شده بود حذف شده و فقط یک بار تکرار می شود.

1. تعریف متد flip

این متد خروجی ندارد و از نوع void می باشد. وظیفه این متد برعکس کردن ترتیب اضلاع در یک لایه پس از استفاده از متد order می باشد. یعنی یک لایه پس از استخراج ابتدا بایستی توسط متد order مرتب شود و پس از آن در صورت نیاز توسط متد flip ترتیب آن بر عکس می شود . شکل های زیر به خوبی عملکرد این متد را نشان می دهند. اگر فرض کنیم لایه S توسط متد extend استخراج شده باشد آنگاه اعمال متد های order و flip به ترتیب اضلاع این لایه را اینگونه مرتب خواهند کرد :

S:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 10 | 9 | 8 | 7 | 6 | 5 | 4 |

S.order():

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

S.flip() :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

1. عملکرد دستور flip

در اینجا ابتدا یک متغیر از نوع surface به اسم N تعریف می شود. سپس ابعاد N بر اساس تعداد اضلاع موجود در surface اصلی تعیین می شود. آن گاه در داخل حلقه به ترتیب از درایه آخر آرایه های مربوطه برداشته می شود و به ابتدای آن آرایه ها ریخته می شود. این روند تا انتها پیگیری می شود به گونه ای که آخرین درایه ای که در N.E قرار می گیرد ، اولین درایه از آرایه E می باشد. برای المان ها نیز به همین صورت. در انتها نیز خود لایه برابر با متغیر N قرار می گیرد.

1. تعریف متد clear

در این بخش تمامی درایه های وکتورهای شئ surface پاک می شود.