

8 月 18 日 笔记

作者: [heyang5188](#)

原文链接: <https://ld246.com/article/1502335646118>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)


```

lt;
</span></span><span class="highlight-line"><span class="highlight-cl"> Screen(pos ht,
os wd, char c) :height(ht), width(wd),
</span></span><span class="highlight-line"><span class="highlight-cl"> contents(ht * w
, c) {}
</span></span><span class="highlight-line"><span class="highlight-cl"> char get() const
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">     return conten
s[cursor];
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> inline char get(
os ht, pos wd) const;
</span></span><span class="highlight-line"><span class="highlight-cl"> Screen &
move(pos r, pos c);
</span></span><span class="highlight-line"><span class="highlight-cl"> public:
</span></span><span class="highlight-line"><span class="highlight-cl"> Screen &
t(char);
</span></span><span class="highlight-line"><span class="highlight-cl"> Screen &
t(pos, pos, char);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> private:
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> pos cursor = 0;
</span></span><span class="highlight-line"><span class="highlight-cl"> pos height = 0 ,
width = 0;
</span></span><span class="highlight-line"><span class="highlight-cl"> std::string cont
nts;
</span></span><span class="highlight-line"><span class="highlight-cl">};
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> inline Screen &a
p;Screen::set(char c)
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> contents[cursor]
= c;
</span></span><span class="highlight-line"><span class="highlight-cl"> return *this;
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl"> inline Screen &a
p;Screen::set(pos r, pos col, char ch)
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> contents[r*width
+ col] = ch;
</span></span><span class="highlight-line"><span class="highlight-cl"> return *this;
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl"> using namespace
td;
</span></span><span class="highlight-line"><span class="highlight-cl"> int main()
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> Screen myScre
n(5, 5, 'X');
</span></span><span class="highlight-line"><span class="highlight-cl"> myScreen.move
4, 0).set('#');
</span></span><span class="highlight-line"><span class="highlight-cl"> cout &&
l;
</span></span><span class="highlight-line"><span class="highlight-cl">

```

```

</span> </span> <span class="highlight-line"> <span class="highlight-cl"> system("pause")
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> return 0;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> #include "stdafx.h

</span> </span> <span class="highlight-line"> <span class="highlight-cl"> #include &lt;iostr
am&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> #include &lt;strin
&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> using namespace
td;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> class Screen {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> unsigned height
= 0, width = 0;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> unsigned cursor
= 0;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> string contents;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> public:
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Screen() = defa
lt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Screen(unsigne
ht, unsigned wd) :height(ht), width(wd), contents(ht*wd, ' ') {}
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Screen(unsigne
ht, unsigned wd, char c) :height(ht), width(wd), contents(ht*wd, c) {}
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Screen&
;
ove(unsigned r, unsigned c)
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> cursor = r*w
th + c;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> return *this;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Screen&
; s
t(char ch)
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> contents[cur
or] = ch;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> return *this;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Screen&
; s
t(unsigned r, unsigned c, char ch)
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> contents[r*w
dth + c] = ch;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> return *this;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Screen&
; d
splay()
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> cout &lt;&lt;
ontents;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> return *this;

```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">};
</span></span><span class="highlight-line"><span class="highlight-cl">int main()
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">    Screen myScreen(5, 5, 'X');
</span></span><span class="highlight-line"><span class="highlight-cl">    myScreen.move(4, 0).set('#').display();
</span></span><span class="highlight-line"><span class="highlight-cl">    cout &&endl;
</span></span><span class="highlight-line"><span class="highlight-cl">    myScreen.display(); //如果取消&符号，上一排输出的是副本，然后这就是新的就没有#
</span></span><span class="highlight-line"><span class="highlight-cl">    system("pause")

</span></span><span class="highlight-line"><span class="highlight-cl">    return 0;
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
```