# 项目目标

1.完成预留地规划方案生成软件

2.掌握如何实现界面

3.掌握继承派生、虚拟和多态

4.掌握输入输出流的编写，生成文件储存方案规划结果。

5.学会对自己的软件写功能说明

6.学会对自己的函数进行逐个测试

7.实现团队合作、分工完成任务

# 项目内容

**1.要求：**

实现预留地规划方案生成软件。要求有界面，生成文件储存方案规划结果。需要对自己的软件写功能说明，对自己的函数进行测试。

**2.功能：**

根据用户输入的场地面积和总预算，若预算充足则输出方案的草图，在草图旁边输出合理方案的具体内容，并在系统内部将方案生成的结果写入文件中；预算不足则提示信息，为何无法生成方案。

如果本次输入数据与上一次方案的输入数据相同时（即用户多次点击生成键），结果将不会被重复写入文件。

**3.使用流程：**

输入：

①场地的面积，输入的范围为：0 ~ 99999999.999999；

②预计的预算，输入的范围为：0 ~ 99999999.999999；

③是否将未利用部分用草坪填满，用户在单选框中选择；

输出：

界面上用户只需要按要求输入数据，点击生成方案按钮，就能将内部算法函数执行的返回结果显示到界面上。

**4.类的介绍**

**Shape类：**

**1.路径：**

定义于 /Headers/base.h 与 /Sources/base.cpp，主要内容均写于base.h。

**2.描述：**

包含了纯虚函数，是抽象类，是接下来所有几何形状的父类。

定义了纯虚方法area, 其中const关键字表明了这个函数不能修改其数据成员且可以被常量对象访问, =0表示了这个函数是纯虚函数。

因为价格基本都是基于面积，所以定义这个纯虚函数要求之后的几何形状类都需要计算面积。

﻿**Circular类：**

**1.路径：**

成员变量和函数定义于﻿circular.h中，函数实现于﻿circular.cpp中。

**2.描述**

该类继承了Shape类，为圆形类，用于表示游泳池；

成员变量：radius：存储了圆形的半径，类型为double。

函数：

area：返回一个double类型的值，无参数，利用圆形面积公式 S = π\* r^2 ，计算圆形的面积并返回；

set\_radius：返回类型为void，有一个double类型的参数，修改类的成员变量radius；

get\_perimeter：返回一个double类型的值，无参数，利用圆形周长公式C = 2 \* π\* r，计算圆形的周长并返回；

get\_radius：返回一个double类型的值，无参数，返回半径即成员变量radius的值。

﻿**Rectangle类：**

**1.路径**

成员变量和函数定义于﻿rectangle.h中，函数实现于﻿rectangle.cpp中。

**2.描述**

该类继承了Shape类，为矩形类，用于表示休息区。

成员变量：

length：存储矩形的长，类型为double；

width：存储矩形的宽，类型为double。

函数：

﻿area：返回一个double类型的值，无参数，利用矩形面积公式 S = length \* width，计算矩形的面积并返回；

﻿set\_attribute：返回类型为void，两个double类型的参数，用于修改类的成员变量length和 width, 第一个参数对应修改length 第二个参数对应修改 width；

﻿get\_perimeter：返回一个double类型的值，无参数，利用矩形周长公式 C = 2 \* (length + width)，计算矩形的周长并返回；

get\_length：返回一个double类型的值，无参数，返回矩形的长即成员变量length的值；

get\_width：返回一个double类型的值，无参数，返回矩形的宽即成员变量width的值。

**Trapezium类：**

**1.路径**

1、成员变量和函数定义于﻿trapezium.h中；

2、函数实现于﻿trapezium.cpp中。

**2.描述**

该类继承了Shape类，为梯形类，且梯形为等腰梯形，且较长的下底的底角为45度，用于表示休息区。

成员变量：

upLength：存储梯形的上底，类型为double ；

downLength：存储梯形的下底，类型为double ；

height：存储梯形的高，类型为double 。

函数：

area：返回一个double类型的值，无参数，利用梯形面积公式 S = 1/ 2 \* height \* (upLength + downLength)，计算梯形的面积并返回；

set\_attribute：返回类型为void，一个double类型的参数，用于修改类的成员变量downLength；

get\_perimeter：返回一个double类型的值，无参数，利用等腰梯形周长公式 C =上底 + 下底 + 2 \* 腰长，计算梯形的周长并返回；

get\_downLength：返回一个double类型的值，无参数，返回梯形的下底即成员变量downLength的值；

get\_upLength：返回一个double类型的值，无参数，返回梯形的上底即成员变量upLength的值；

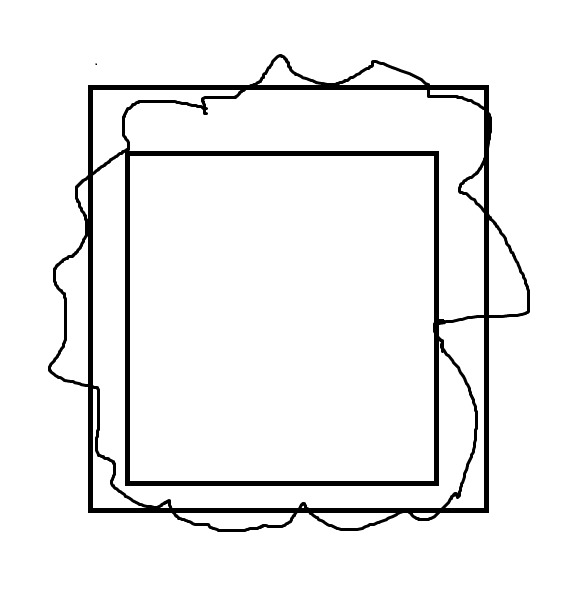
get\_height：返回一个double类型的值，无参数，返回梯形的高即成员变量height的值。

**5.核心算法详细介绍**

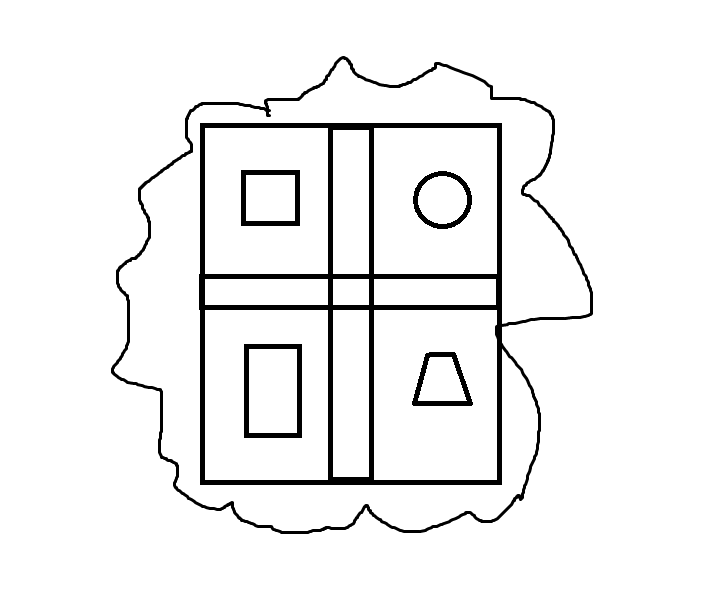
**主要算法包括两个逻辑函数，calculate和search**

第一，将预留地做合理化处理。用户输入总场地面积开方向下取整，再去掉得到边长的1/20以减小误差实现场地的最大利用（图形不规则，内部画方可能有边角取不到，消去一点以合理化）将不规则场地化作规则正方形。

如下图所示，场地不规则，外方形为直接开方所得，内方形为经算法处理实际可利用面积。



第二，search函数寻找最合理方案。首先将场地化为四块，在这四块分别建设不同设施，有圆形泳池（外围有栅栏）、矩形更衣室、方形超市和梯形休息区，中间过道（两边均有栅栏）默认为三米宽；再利用第一步中我们得到可利用边长，用可利用边长千分之一作为计数器，四部分边长（或某参数）从零开始寻找合理预算能建成的最大面积。以千分之一作为计数器是为了减少计算时间的同时保证精度，若以1米为计量单位对10米场地计算误差过大，浪费空间过大；对10000米场地来说计算耗时过多。选取合理计数器以边长的千分之一为单位。下图为方案草图



在这一步如果最终计算出合理边长不足可以利用边长的1/4（即分块变长的1/2），说明规划完成方案场地（过道加设施）利用率低于50%，则输出预算不足无法形成方案。

第三，可处理未利用场地，以实现利用最大化。外围因不规则截去场地设计方案为种树，绿色森林环绕着整个娱乐场；内部（圆形、梯形、矩形边角）未利用场地由客户选择是否以草坪覆盖，无论是否填充都可以给客户输出最合理的方案和各部分支出预算。当然，覆盖草坪花费可能会导致其他部分预算减少，我们通过压缩草地花费、总场地四部分平均减少预算，以尽可能减小铺盖草坪对其他部分场地建设的影响。

最后，以字符串输出各类信息：各部分单价，预计方案造价，总预算以及场地利用情况。

**算法参数列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 变量 | 类型 | 解释 | 其他注释 |
| area | double | 场地面积 | 传参、用户输入 |
| budget | double | 总预算金额 | 传参、用户输入 |
| use\_budget | double | 终预算 | 引用值 |
| side | double | 边长 | 引用值，各形状最终参数 |
| with\_grass | bool | 是否要草坪 | 1则草地填满剩余区域，0反之 |
|  |  |  | 以上为calculate函数所需参数 |
|  |  |  |  |
| square\_side | double | 外边长 | 最大可利用方形边长 |
| count | double | 计数器 | 千分之一外边长，扩展最小单位 |
| tree\_budget | double | 树所需预算 | 必须满足，无法减少budget |
| result | string | 最终方案 | 为文字描述 |
|  |  |  | 以上为calculate函数内部定义参数 |
|  |  |  |  |
| area | 同上 |  |  |
| budget | 同上 |  |  |
| square\_side | 同上 |  |  |
| count | 同上 |  |  |
| tree\_budget | 同上 |  |  |
| use\_budget | 同上 |  | 依然作为引用 |
| side | 同上 |  | 依然作为引用 |
|  |  |  | 以上为search函数所需参数 |
|  |  |  |  |
| square\_shop | Rectangle | 方形超市 |  |
| rectangle\_locker | Rectangle | 矩形更衣室 |  |
| circular\_poor | Circular | 圆形泳池 |  |
| trapezoid\_rest | Trapezoid | 梯形休息区 |  |
| result | string | 最终方案 | 为文字描述 |
|  |  |  |  |
| shop\_area | double | 超市面积 | 用于计算造价 |
| locker\_area | double | 更衣室面积 | 用于计算造价 |
| poor\_area | double | 泳池面积 | 用于计算造价 |
| fence\_length | double | 篱笆长度 | 用于计算造价 |
| reat\_area | double | 休息区面积 | 用于计算造价 |
|  |  |  |  |
| shop\_budget | double | 超市预算 |  |
| locker\_budget | double | 更衣室预算 |  |
| poor\_budget | double | 泳池预算 |  |
| fence\_budget | double | 篱笆预算 |  |
| reat\_budget | double | 休息区预算 |  |
|  |  |  | 以上为search函数自定义参数 |
|  |  |  |  |
| grass\_price |  | 草造价 | All |
| tree\_price |  | 树造价 | 元/平方米 |
| shop\_price |  | 超市造价 | 元/平方米 |
| locker\_price |  | 更衣室造价 | 元/平方米 |
| poor\_price |  | 泳池造价 | 元/平方米 |
| fence\_price |  | 篱笆造价 | 元/米 |
| reat\_price |  | 休息区造价 | 元/平方米 |

**6.分工协作**

蒋志文 界面

张宜放 算法

杨震东 类

姚江 测试

# 项目优势

使用qt库，自主设计的算法，实现继承派生、虚拟、多态，文件目录清晰，生成并保存输出文件以及友好的界面。