**课程设计任务书**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目名称 | 面向对象设计+图形化界面设计+数据库编程 | | | 实验成绩 |  |
| 实验者 |  | 专业班级 |  | 实验日期 | 2022-06-04 |
| 第一部分：实验分析与设计   1. 实验目标   掌握Tkinter窗口创建、坐标管理器  掌握标签、按钮、输入框、列表框、画布等Tkinter组件的使用   1. 实验内容   （一）图形化界面设计要求：  **综合运用所学tkinter进行图形界面化设计，**并进行合理布局，完成相应功能。  **1.根据功能需求所选控件，要求至少要包含>=3的不同控件。**  **2.控件能够按照功能需求进行合理布局，控件可以随着窗体的大小灵活缩放。**  **3.事件驱动与回调机制的设计，能够适用功能需求，满足功能设计需要。**  **占期末成绩60%**  **评分标准：**  **功能复杂程度30%**  **控件应用程度20%**  **界面布局合理10%**  **事件驱动与回调机制设计20%**  **功能实现状况10%**  **生成exe程序10%**  **（二）Mysql数据库的设计**  **占期末成绩40%**  **自拟应用场景，在Mysql数据库中**  **（1）创建数据库 （5分）**  create database room;  use room;  **（2）创建数据表：至少要两张表可以进行多表连接（5分）**  DROP TABLE IF EXISTS `customer\_info`;  CREATE TABLE `customer\_info` (  `customer\_id` int(0) NOT NULL,  `room\_id` int(0) NOT NULL,  `check\_in\_date` date NOT NULL,  `check\_out\_date` date NULL DEFAULT NULL,  `total\_cost` decimal(10, 2) NULL DEFAULT NULL,  `username` varchar(50) NOT NULL,  PRIMARY KEY (`customer\_id`),  INDEX `room\_id`(`room\_id`),  CONSTRAINT `customer\_info\_ibfk\_1` FOREIGN KEY (`room\_id`) REFERENCES `room\_info` (`room\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT  );  INSERT INTO `customer\_info` VALUES (1, 1, '2023-06-21', '2023-06-23', 200.00, '张三');  INSERT INTO `customer\_info` VALUES (2, 2, '2023-06-21', '2023-06-25', 300.00, '王五');  INSERT INTO `customer\_info` VALUES (3, 3, '2023-06-22', '2023-06-26', 400.00, '刘麻子');  INSERT INTO `customer\_info` VALUES (4, 4, '2023-06-23', '2023-06-30', 900.00, '小猫');  INSERT INTO `customer\_info` VALUES (5, 5, '2023-06-24', '2023-06-29', 750.00, '小月月');  INSERT INTO `customer\_info` VALUES (6, 6, '2023-06-25', '2023-06-28', 540.00, '洋洋');  INSERT INTO `customer\_info` VALUES (7, 7, '2023-06-26', '2023-06-27', 90.00, '沸羊羊');  INSERT INTO `customer\_info` VALUES (8, 8, '2023-06-27', '2023-06-29', 240.00, '舔狗');  DROP TABLE IF EXISTS `room\_info`;  CREATE TABLE `room\_info` (  `room\_id` int(0) NOT NULL,  `type\_id` int(0) NOT NULL,  `room\_number` varchar(20) NOT NULL,  `status` varchar(10) NOT NULL DEFAULT 'Vacant',  PRIMARY KEY (`room\_id`),  UNIQUE INDEX `room\_number`(`room\_number`),  CONSTRAINT `room\_info\_ibfk\_1` FOREIGN KEY (`type\_id`) REFERENCES `room\_type` (`type\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT  );  INSERT INTO `room\_info` VALUES (1, 1, '101', 'Vacant');  INSERT INTO `room\_info` VALUES (2, 2, '102', 'Vacant');  INSERT INTO `room\_info` VALUES (3, 3, '103', 'Vacant');  INSERT INTO `room\_info` VALUES (4, 4, '104', 'Vacant');  INSERT INTO `room\_info` VALUES (5, 5, '105', 'Vacant');  INSERT INTO `room\_info` VALUES (6, 6, '201', 'Vacant');  INSERT INTO `room\_info` VALUES (7, 7, '202', 'Vacant');  INSERT INTO `room\_info` VALUES (8, 8, '203', 'Vacant');  DROP TABLE IF EXISTS `room\_price`;  CREATE TABLE `room\_price` (  `price\_id` int(0) NOT NULL,  `type\_id` int(0) NOT NULL,  `start\_date` date NOT NULL,  `end\_date` date NOT NULL,  `price` decimal(10, 2) NOT NULL,  PRIMARY KEY (`price\_id`),  CONSTRAINT `room\_price\_ibfk\_1` FOREIGN KEY (`type\_id`) REFERENCES `room\_type` (`type\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT  );  INSERT INTO `room\_price` VALUES (1, 1, '2023-06-21', '2023-06-30', 100.00);  INSERT INTO `room\_price` VALUES (2, 2, '2023-06-21', '2023-06-30', 150.00);  INSERT INTO `room\_price` VALUES (3, 3, '2023-06-21', '2023-06-30', 200.00);  INSERT INTO `room\_price` VALUES (4, 4, '2023-06-21', '2023-06-30', 300.00);  INSERT INTO `room\_price` VALUES (5, 5, '2023-06-21', '2023-06-30', 250.00);  INSERT INTO `room\_price` VALUES (6, 6, '2023-06-21', '2023-06-30', 180.00);  INSERT INTO `room\_price` VALUES (7, 7, '2023-06-21', '2023-06-30', 90.00);  INSERT INTO `room\_price` VALUES (8, 8, '2023-06-21', '2023-06-30', 120.00);  DROP TABLE IF EXISTS `room\_type`;  CREATE TABLE `room\_type` (  `type\_id` int(0) NOT NULL,  `type\_name` varchar(50) NOT NULL,  `price` decimal(10, 2) NOT NULL,  PRIMARY KEY (`type\_id`)  );  INSERT INTO `room\_type` VALUES (1, 'Standard', 100.00);  INSERT INTO `room\_type` VALUES (2, 'Deluxe', 150.00);  INSERT INTO `room\_type` VALUES (3, 'Suite', 200.00);  INSERT INTO `room\_type` VALUES (4, 'VIP', 300.00);  INSERT INTO `room\_type` VALUES (5, 'Family', 250.00);  INSERT INTO `room\_type` VALUES (6, 'Executive', 180.00);  INSERT INTO `room\_type` VALUES (7, 'Single', 90.00);  INSERT INTO `room\_type` VALUES (8, 'Double', 120.00);    **（3）对表中数据进行增加、修改、删除、查询操作（有多表连接的查询）（20分）**  **插入：**  INSERT INTO `room\_type` VALUES (9, 'Premium', 400.00);    修改：  UPDATE `room\_price` SET `price` = 250.00 WHERE `price\_id` = 5;    删除：  DELETE FROM `customer\_info` WHERE `customer\_id` = 2;    多表查询：  SELECT ci.customer\_id, ci.username, ri.room\_number, rt.type\_name  FROM `customer\_info` ci  JOIN `room\_info` ri ON ci.room\_id = ri.room\_id  JOIN `room\_type` rt ON ri.type\_id = rt.type\_id;    **（4）能够结合面向对象程序设计，图形界面设计，组件与数据库联动实现数据的增删改查。（10分）**  三、主要仪器设备及软件环境  1.PC机  2.jupyter notebook | | | | | |
| 第二部分：实验调试与结果分析   1. 调试过程（源程序等，包括调试方法描述、实验数据记录、实验过程发现的问题等）   （一）图形界面设计部分  1.功能描述   1. 添加入住信息：输入客户ID、房间ID、入住日期、退房日期、总费用和客户姓名，点击添加按钮将入住信息添加到数据库。 2. 查询入住信息：输入客户ID，点击查询按钮查询并显示该客户的入住信息。 3. 更新入住信息：输入客户ID、房间ID、入住日期、退房日期、总费用和客户姓名，点击更新按钮更新该客户的入住信息。 4. 删除入住信息：输入客户ID，点击删除按钮删除该客户的入住信息。 5. 查询空房间：点击查询按钮查询并显示所有空房间的信息。 6. 按照时间查询入住信息：输入起始日期和终止日期，点击查询按钮查询并显示在该时间范围内入住的所有客户的入住信息。   2.使用控件详细介绍  Label：用于显示文本标签。  Entry：用于输入文本。  Button：用于触发事件。  Text：用于显示多行文本。  3.界面布局介绍  使用grid布局，将控件按照行列的方式进行排列。  4.事件驱动与回调机制设计思路  通过给按钮绑定相应的事件处理函数，当按钮被点击时，会触发对应的事件处理函数  5.有无自学部分  自学了python的thinker的回调函数，菜单栏，数据get。  6.其他需要说明的部分  （二）数据库编程部分  1.应用场景描述  该数据库用于存储客房的入住信息，包括客户ID、房间ID、入住日期、退房日期、总费用和客户姓名。  用来管理酒店客房入住信息的增删改查操作，提高了效率，减少出错。  2.创建数据库  创建名为"room"的数据库  3.创建数据表  创建名为"customer\_info"的表，包含字段：customer\_id、room\_id、check\_in\_date、check\_out\_date、total\_cost和username。  创建名为"room\_info"的表，包含字段：room\_id、type\_id、room\_number和status。  创建名为"room\_price"的表，包含字段：price\_id、type\_id、start\_date、end\_date和price。  创建名为"room\_type"的表，包含字段：type\_id和type\_name。  4.对数据库中数据进行增加、删除、修改、查询  在上面。  5.结合图形界面设计、面向对象设计、对数据库中数据进行增删改查。  详细件结果分析   1. 实验结果及分析（运行结果截屏）   程序界面：    添加用户入住信息：    查询用户：    更新用户，更换房间    删除用户    查询空房间：    按照时间查询全部用户入住记录    按照姓名查询全部记录     1. 实验小结、建议及体会   这个程序设计实践的实验课程主要包括两个部分：图形化界面设计和数据库编程。  在图形化界面设计部分，要求使用Tkinter库进行窗口创建和控件的使用。具体要求包括选择至少3种不同的控件，并进行合理布局，使得控件能够随窗体大小灵活缩放。还要设计事件驱动和回调机制，满足功能需求，并实现功能复杂程度、控件应用程度、界面布局合理性、事件驱动与回调机制设计和功能实现状况等方面的评分标准。  在数据库编程部分，要求使用MySQL数据库进行设计。要求自拟应用场景，并创建数据库和数据表，进行增加、修改、删除、查询操作，包括多表连接的查询。还要结合面向对象程序设计、图形界面设计和数据库联动，实现数据的增删改查。  根据实验结果截图，程序界面设计看起来布局合理，并且能够实现添加、查询、更新和删除用户入住信息的功能。同时，还能查询空房间和按照时间和姓名查询全部用户入住记录的功能。  总的来说，这个实验课程是一个综合性的设计实践，涉及到图形化界面设计和数据库编程的综合应用。通过完成该实验，可以加深对Tkinter库和MySQL数据库的理解和应用，同时提升面向对象程序设计能力和数据库操作能力。建议在实验过程中注重细节，确保程序的稳定性和功能的完整性。 | | | | | |