

患者健康记录管理系统设计要求

这个系统设计考察Python重要知识点包括：

- 面向对象编程
- 文件操作
- 异常处理
- 数据结构（列表和字典）
- 模块化设计
- 用户交互
- 数据验证和处理

提交时间：2024年12月1日 24:00 之前

提交路径：TronClass系统提交，不支持邮件提交

一、系统架构（文件目录）

```
health_system/
├── data/
│   └── patient_records.txt      # 患者数据文件
├── models/
│   ├── __init__.py
│   └── patient.py              # 患者模型类
├── utils/
│   ├── __init__.py
│   ├── data_manager.py        # 数据管理类
│   └── health_analyzer.py     # 健康分析类
└── main.py                    # 主程序
```

二、数据格式规范

患者记录文件格式

- 提供了一个包含100条**AI生成的仿真患者数据库**[1]文件 `patient_records.txt`，新增、更新和删除患者需要记录到这个文件中

- 每行一条记录，字段用制表符(\t)分隔
- 字段顺序：

```
patient_id    name    age    gender    height    weight    blood_pressure    blood_sugar
```

- 数据类型和约束：
 - patient_id: 字符串，以'P'开头
 - name: 字符串
 - age: 整数 (0-120)
 - gender: 字符串 ('男'或'女')
 - height: 浮点数 (50-250cm)
 - weight: 浮点数 (20-200kg)
 - blood_pressure: 字符串 (格式：收缩压/舒张压)
 - blood_sugar: 浮点数
 - cholesterol: 浮点数
 - check_date: 字符串 (格式：YYYY-MM-DD)

[1] 已有100条患者记录数据均为AI生成，如有重名纯属巧合。

三、功能要求

功能类别	功能项目	具体要求
基本功能	患者信息管理	1. 查看患者信息 2. 添加新患者 3. 更新患者信息 4. 删除患者 5. 生成健康报告
数据验证	输入验证	1. 所有输入数据必须进行有效性验证 2. 数值范围检查 3. 格式检查 4. ID唯一性检查
健康分析	健康指标评估	1. BMI计算和分类 2. 血压状况分析 3. 血糖水平评估 4. 胆固醇水平评估
异常处理	异常情况管理	1. 文件操作异常 2. 数据格式异常 3. 用户输入异常 4. 数据验证异常
数据持久化	文件操作	1. 自动加载数据文件 2. 数据修改后自动保存 3. 文件不存在时自动创建 4. 文件导出和存储

三、模块设计要求

1. 患者模型 (models/patient.py)

Patient 类

```
class Patient:
    def __init__(self, patient_id, name, age, gender, height, weight,
                  blood_pressure, blood_sugar, cholesterol, check_date)
        """
        功能：初始化患者对象
        参数：所有患者基本信息字段
        异常：无
        """

    def validate(self)
        """
        功能：验证患者数据的有效性
        返回：bool - 验证是否通过
        异常：ValueError - 数据验证失败时抛出，包含具体错误信息
        """

    def to_string(self)
        """
        功能：将患者信息转换为制表符分隔的字符串
        返回：str - 格式化的字符串
        """

    def __str__(self)
        """
        功能：返回患者信息的可读字符串表示
        返回：str - 格式化的字符串
        """
```

2. 数据管理 (utils/data_manager.py)

DataManager 类

```
class DataManager:
    def __init__(self, file_path)
        """
        功能: 初始化数据管理器
        参数: file_path (str) - 数据文件路径
        """

    def load_data(self)
        """
        功能: 从文件加载患者数据
        异常:
            FileNotFoundError - 文件不存在
            Exception - 其他加载错误
        """

    def save_data(self)
        """
        功能: 保存患者数据到文件
        异常: Exception - 保存失败
        """

    def add_patient(self, patient)
        """
        功能: 添加新患者
        参数: patient (Patient) - 患者对象
        异常:
            ValueError - 患者ID已存在
            ValueError - 数据验证失败
        """

    def get_patient(self, patient_id)
        """
        功能: 获取指定ID的患者信息
        参数: patient_id (str) - 患者ID
        返回: Patient对象或None
        """
```

```
def update_patient(self, patient)
"""
功能：更新患者信息
参数：patient (Patient) - 更新后的患者对象
异常：
    ValueError - 患者不存在
    ValueError - 数据验证失败
"""

def delete_patient(self, patient_id)
"""
功能：删除患者记录
参数：patient_id (str) - 患者ID
异常：ValueError - 患者不存在
"""

def get_all_patients(self)
"""
功能：获取所有患者列表
返回：list[Patient] - 患者对象列表
"""
```

3. 健康分析 (utils/health_analyzer.py)

HealthAnalyzer 类

```
class HealthAnalyzer:
    @staticmethod
    def calculate_bmi(height, weight)
    """
    功能：计算BMI指数
    参数：
        height (float) - 身高(cm)
        weight (float) - 体重(kg)
    返回：tuple(float, str) - (BMI值, BMI分类)
    """

    @staticmethod
    def analyze_blood_pressure(bp_str)
    """
    功能：分析血压状况
    参数：bp_str (str) - 血压字符串(格式：收缩压/舒张压)
    返回：str - 血压状况描述
    """

    def generate_health_report(self, patient)
    """
    功能：生成患者健康报告，并以"[patient_id]_report.txt"文件格式存储在"health_system/"主文件夹下
    参数：patient (Patient) - 患者对象
    返回：str - 格式化的健康报告
    """
```

4. 主程序 (main.py)

HealthSystem 类

```
class HealthSystem:
    def __init__(self)
        """
        功能：初始化健康记录管理系统
        """

    def display_menu(self)
        """
        功能：显示系统主菜单
        """

    def input_patient_data(self)
        """
        功能：交互式输入患者数据
        返回：Patient - 新的患者对象
        异常：ValueError - 输入数据无效
        """

    def run(self)
        """
        功能：运行系统主循环
        """
```

五、评分标准

评分维度	评分项目	评分内容	分值
一、基础架构	项目结构	1. 完整的模块化结构 2. 合理的文件组织 3. 正确的包导入关系	5
	代码规范	1. 代码风格符合PEP 8 2. 完整的Docstring 3. 适当的代码注释	15
二、核心功能	Patient类	1. 属性定义完整 2. 数据验证功能完整 3. 字符串转换方法实现	10

评分维度	评分项目	评分内容	分值
	DataManager类	1. 文件读写功能 2. CRUD操作完整性 3. 异常处理完善	15
	HealthAnalyzer类	1. BMI计算功能 2. 血压分析功能 3. 血糖分析功能 4. 胆固醇分析功能 5. 健康报告生成	15
三、 数据处理	数据验证	1. 输入格式验证 2. 数值范围检查 3. ID唯一性检查 4. 日期格式验证	10
	异常处理	1. 文件操作异常 2. 数据格式异常 3. 用户输入异常 4. 数据验证异常	10
四、 系统功能	用户界面	1. 菜单显示清晰 2. 操作提示完整 3. 错误提示友好	10
	功能完整性	1. 查看功能实现 2. 添加功能实现 3. 更新功能实现 4. 删除功能实现 5. 报告生成功能	10