南京审计大学

程序设计课程设计报告

组　　别：计算机二班第三组

课　　题：校园动物管理系统

课 程 名 称：程序设计课程设计

指 导 教 师：陈勇

学　　院：计算机学院

学　　校：南京审计大学

日　　期：2023 年 8 月 16 日

校园动物管理系统

作者：

学生姓名	分工
方卓澄	主菜单设计与实现
黄晨宇	文件读写模块
金太宇	数据查询模块
陈　彪	数据修改模块

目录

Chapter 1

绪论

1.1 系统背景

校园动物管理系统的系统背景在于科学、规范、高效地管理校园内的各类动物，旨在保护校园内的动物安全，促进人与动物和谐共处。通过建立动物信息数据库，对每只动物进行记录，包括动物的种类、性别、年龄、主要栖息地等信息，以便于统计和管理。保障校园内动物的安全和福利，同时推动人与动物和谐共处，为构建文明和谐的校园做出积极贡献。本系统基于 C 语言开发，旨在为学校管理不同种类的动物提供一种方便、快捷的管理方式。该系统可以帮助轻松管理动物信息，并支持动物信息的增删、查询等功能。

1.2 系统解决的主要问题

该系统能解决以下几个主要问题：

1. 登记管理：对每只校园动物进行信息登记管理，包括动物名字、照片、数量、品种等等数据。
2. 定时喂养：设定不同时间段、不同区域的喂养计划，针对流浪动物设定相应的食物、量和时间。
3. 健康监护：提供动物兽医服务，实行定期检查，发现问题及更换治疗，同时，实现动物健康档案管理。
4. 信息查询：允许管理员查询动物信息，查看动物的生长情况、疾病状态等等。

1.3 设计目的

该系统的设计目的在于为高校提供一种有效地管控与管理校园动物的方式。此外，通过采用先进技术手段，推行动物福利保障工作，营造校园内和谐的气氛。同时，该系统也可以方便前来管理者对流浪动物进行关注和宣传，提高公共关注度，并加强大众保护动物意识。最后，该系统能够增强爱心，推动志愿者服务，实现公益性的覆盖和管理。

Chapter 2

需求分析与总体设计

学校动物管理系统需要满足以下需求：

- **动物信息管理**：对不同种类动物的信息进行增删改查等操作，包括动物种类、年龄、性别、健康状况等。
- **数据存储**：系统可以对动物信息进行存储，以防止数据丢失，并支持在需要时进行数据恢复。
- **界面友好**：系统的界面需要友好、简洁，方便快速了解动物信息和系统操作。

2.1 系统功能

2.1.1 基本功能

- **添加记录**：通过选择菜单选项，可以方便地添加新的动物记录到系统中。可以输入动物的编号、名称、性别、年龄、栖息地和物种信息。
- **删除记录**：提供了按照不同的条件进行删除记录的功能。可以按照动物的编号、名称、年龄、栖息地或物种进行查找并删除对应的记录。
- **修改记录**：同样提供了按照不同的条件进行修改记录的功能。可以按照动物的编号、名称、年龄、栖息地或物种进行查找并修改对应的记录。
- **查询记录**：支持按照不同的条件进行查询记录。可以按照动物的编号、名称、年龄、栖息地或物种进行查找，并显示符合条件的记录。

2.1.2 特色功能

- **菜单界面设计**：系统在菜单界面上使用了 ASCII 来呈现一只小猫的图形，增加了用户界面的趣味性和吸引力。
- **外部文件读写**：系统在程序入口处调用了 read() 和 write() 函数，以读取和写入缓冲文件中的数据。这样，即使系统关闭再打开，之前输入的动物记录也能被保留。
- **多级菜单**：系统通过使用嵌套的菜单函数（showmenu2(), showmenu3(), showmenu4()），实现了多级菜单的功能。这样，用户可以根据自己的需求选择不同的操作，提高了系统的灵活性和易用性。

2.2 业务流程

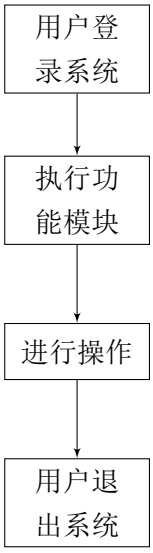


图 2.1: 业务流程

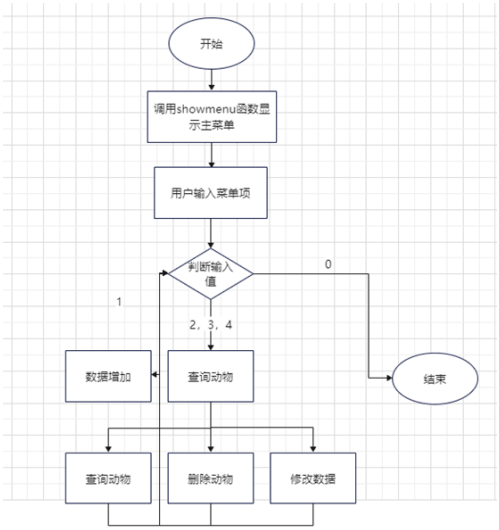
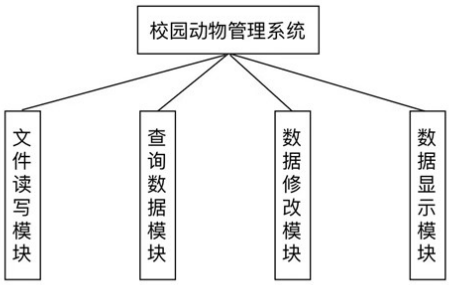


图 2.2: 业务流程

2.3 模块划分及分工

姓名	职责
方桌澄	主菜单设计与实现
黄晨宇	文件读写模块
金太宇	数据查询模块
陈彪	数据修改模块

(a) 模块划分和分工



(b) 模块划分

图 2.3: 模块划分及分工

- 文件读写模块该模块负责处理与文件的读取和写入相关的操作。它包括函数用于打开、关闭和创建文件，以及读取和写入文件中的数据。文件读写模块根据需求支持文本文件格式（txt）。
- 查询数据模块该模块提供查询功能，允许用户根据特定的条件搜索和检索动物管理系统中的数据。它包含函数或方法，用于执行各种类型的查询，如按编号、姓名、年龄、性别、栖息地、物种等查询动物信息。查询数据模块可以根据用户的查询条件返回符合条件的动物信息。
- 数据修改模块该模块负责对动物管理系统中的数据进行修改和更新。它包含函数或方法，用于添加新动物、删除现有动物或修改动物的属性。数据修改模块提供一些接口，以便用户能够方便地输入要修改的动物信息。
- 数据显示模块该模块用于将动物管理系统中的数据显示给用户。它包括函数或方法，用于以用户友好的方式展示动物信息，如列表、表格、图形等形式。数据显示模块提供界面，使用户能

够浏览和查看动物的属性，并提供适当的交互方式。这些模块共同组成了校园动物管理系统的核心功能。每个模块在系统中扮演着不同的角色，通过协同工作，使系统能够实现对动物数据的读取、查询、修改和展示等操作。

Chapter 3

详细设计与实现

3.1 函数接口定义

读写模块：

```
void read();
void write();
```

修改模块：

```
void Add();
void Delete(int n[]);
void Change(int n[]);
```

查询模块：

```
// 输出格式函数声明
void PrintHeader(void);
void PrintRow(const struct Animal* animal);
// 查询函数声明
int* SearchByNum(struct Animal animals[], int animal_num, int target_num);
int* SearchByName(struct Animal animals[], int animal_num, char name[]);
int* SearchByAge(struct Animal animals[], int animal_num, char target_age[]);
int* SearchByHabitat(struct Animal animals[], int animal_num, char target_habitat[]);
int* SearchBySpecies(struct Animal animals[], int animal_num, char target_species[]);
```

3.2 模块内部详细设计

函数 `read()`

函数目的

`read` 函数的目的是从文件中读取动物信息，并将其存储到数组中。

输入参数

无

输出参数

无

返回值

成功读取并存储动物信息时，返回 0。如果文件无法打开或读取失败，函数将打印错误消息并返回 1。

函数流程

```
打开文件 "txt/animal.txt" 以供读取
如果文件无法打开，打印错误消息 "无记录" 并返回 1
定义变量 numLines 用于记录已读取的行数，初始值为 0
使用循环逐行读取文件内容并存储到数组中，直到文件的末尾
使用 fscanf 函数从文件中读取一行动物信息
格式为 \%d \%s \%s \%s \%s \%s
依次将读取的数据存储到 animal[numLines] 结构体中
每次循环结束时，numLines 自增 1
将 numLines 的值赋给全局变量 record\_num，以记录读取的动物记录数
关闭文件
返回 0，表示成功读取并存储动物信息
```

函数 write()

函数目的

write 函数的目的是将动物信息写入文件。

输入参数

无

输出参数

无

返回值

无

函数流程

打开文件 "txt/animal.txt" 以供写入。
使用循环遍历动物数组中的每个动物。
使用 fprintf 函数将动物信息写入文件，格式为 \%d \%s \%s \%s \%s \%s。
关闭文件。

函数 Add()

函数目的

Add 函数的目的是添加新的动物记录。

输入参数

无

输出参数

无

返回值

无

函数流程

打印提示消息 "请输入编号:" 并使用 scanf 函数读取用户输入的编号
将其存储到 animal[record_num].num 中
使用 getchar 函数清除输入缓冲区中的换行符
打印提示消息 "请输入物种:" 并使用 scanf 函数读取用户输入的物种
将其存储到 animal[record_num].species 中
使用 getchar 函数清除输入缓冲区中的换行符
依次重复步骤 3 和 4，分别读取用户输入的姓名、年龄、性别和栖息地
并将它们存储到相应的结构体成员变量中
将全局变量 record_num 自增 1，以表示添加了一条动物记录

函数 Delete(int n[])

函数目的

Delete 函数的目的是根据指定的编号数组，删除对应的动物记录。

输入参数

n[]：要删除的动物编号数组。

输出参数

无

返回值

无

函数流程

使用两层循环：
外层循环遍历指定的编号数组 n[]。
内层循环遍历结构体数组 animal[]。

当找到匹配的编号时，执行以下操作：
使用一个临时变量 k，将其初始化为当前内层循环的索引 j
使用循环将 animal[k] 的值向前推移一位，覆盖掉当前位置的动物记录，直到数组末尾
将全局变量 record_num 自减 1，表示删除了一条动物记录

函数 Change(int* n)

函数目的

Change 函数的目的是根据指定的编号数组，修改对应的动物记录的某些属性。

输入参数

n：要修改的动物编号数组。

输出参数

无

返回值

无

函数流程

定义变量 choice，用于存储用户选择的修改内容
使用两层循环：
外层循环遍历指定的编号数组 n
内层循环遍历结构体数组 animal
当找到匹配的编号时，执行以下操作：
打印提示消息 "请选择你想要改变的内容:\n" 和可供选择的修改内容列表
使用 scanf 函数读取用户选择的修改内容，并将其存储到 choice 变量中
使用 switch 语句根据用户选择执行相应的修改操作：

根据选择的值修改对应的结构体成员变量
释放内存，将编号数组 n 的内存空间释放

函数 SearchByNum(struct Animal animals[], int animal_num, int target_num)

该函数用于按编号查询动物记录，并返回匹配的动物编号数组。

输入参数

animals：结构体数组，表示所有的动物记录。
animal_num：整数，表示总共的动物数量。
target_num：整数，表示目标查询的动物编号。

输出参数

无

返回值

类型：int*（整型指针）
说明：返回一个整型数组，其中存储了匹配的动物编号。

函数流程

调用函数 PrintHeader() 打印表格的表头
初始化计数变量 count 为 0
使用动态内存分配函数 malloc() 分配内存空间，用于存储返回的匹配动物编号的数组
数组的大小为 2 * animal_num * sizeof(int) 字节
使用 memset() 函数将其初始化为 0
使用循环遍历动物数组
如果找到与目标编号匹配的动物记录，则执行以下操作：
    计数变量 count 自增。
    调用函数 PrintRow() 打印整条记录。
    将动物编号存储到返回值数组 return_value 的相应位置。
根据计数变量 count 的值判断是否找到匹配的动物记录：
如果 count 大于 0，则返回动态分配的数组 return_value。
否则，打印提示信息"未找到该动物"，并返回动态分配的数组 return_value。

3.3 核心功能

- 读取动物信息：从文件中读取保存的动物信息，并将其存储到内存中的数据结构中。
- 显示动物信息：将内存中的动物信息以表格的形式显示在控制台上。
- 添加动物信息：通过用户输入，添加新的动物信息，并将其保存到内存中的数据结构和文件中。

- 删除动物信息：根据用户指定的动物编号，删除相应的动物信息，并更新内存中的数据结构和文件。
- 修改动物信息：根据用户指定的动物编号，修改相应的动物信息，并更新内存中的数据结构和文件。
- 查询动物信息：根据用户指定的动物编号，查询并显示匹配的动物信息。
- 保存动物信息：将内存中的动物信息保存到文件中，以便下次程序启动时可以读取。

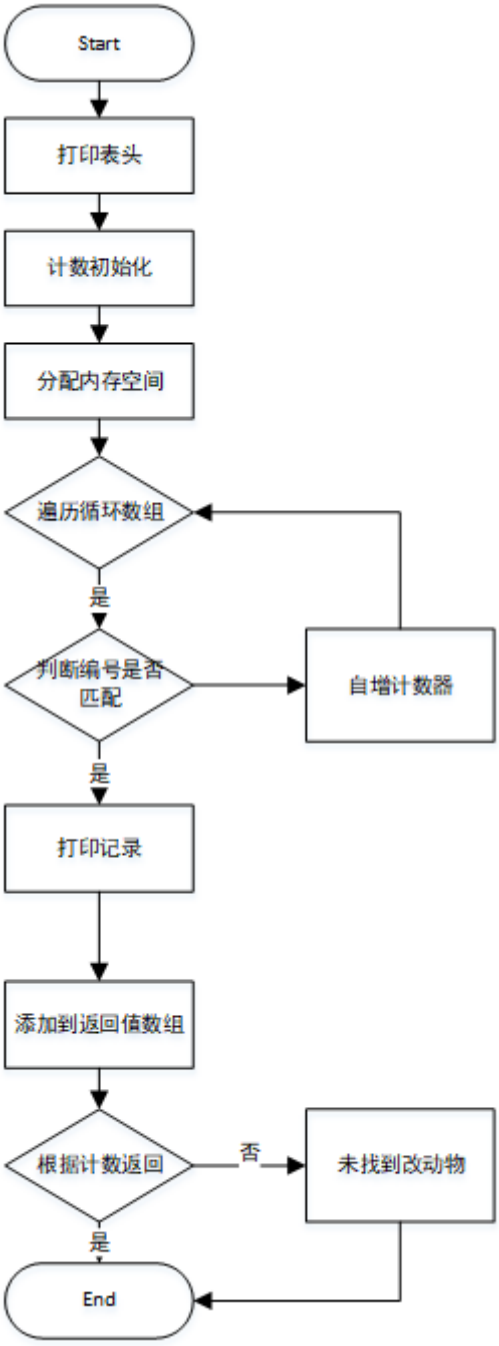
3.4 实现效果

添加记录：选择此选项可以添加新的记录。

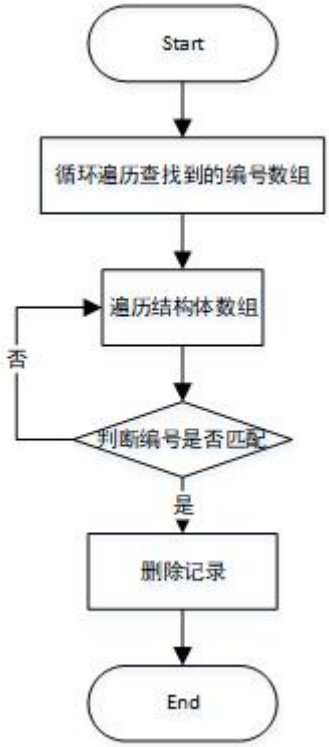
删除记录：选择此选项以删除现有的记录，可以通过跳转查询界面以删除指定记录。

修改记录：选择此选项以修改现有的记录。可以指定要修改的记录编号或关键词进行搜索，然后对记录进行更新.

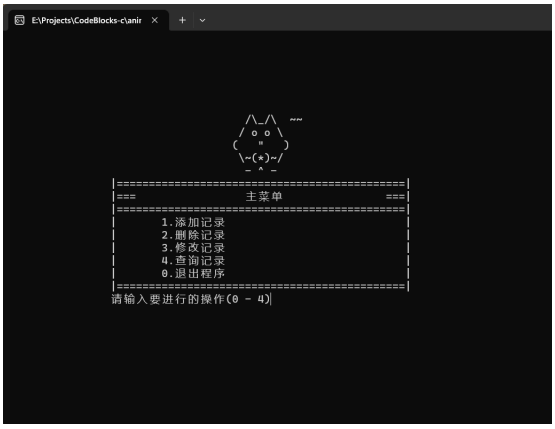
查询记录：选择此选项以查询已有的记录。可以按不同关键词进行查询。



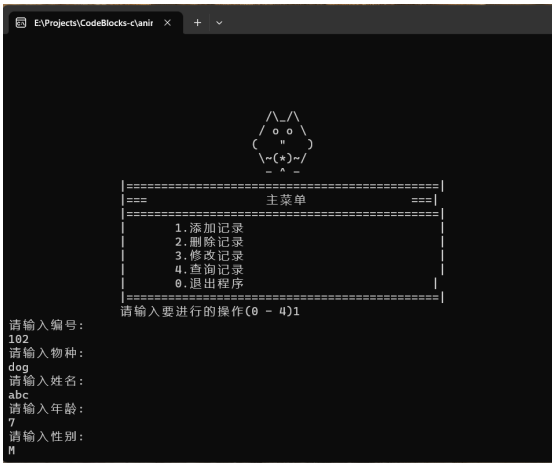
(a) 查询函数流程图



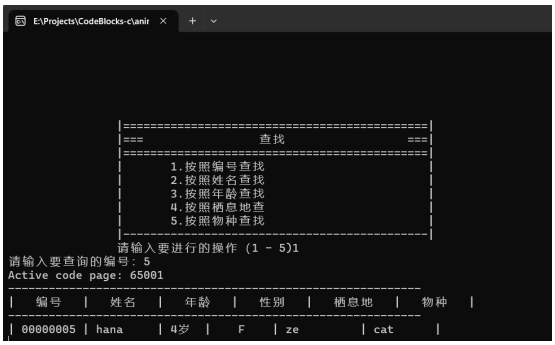
(b) 删除函数流程图



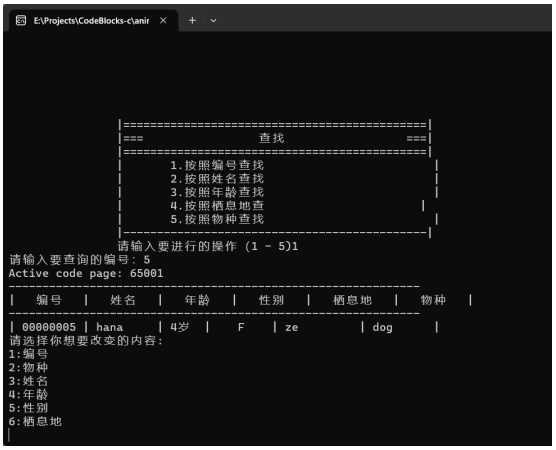
(a) 主菜单



(b) 添加记录



(a) 删除记录



(b) 修改记录

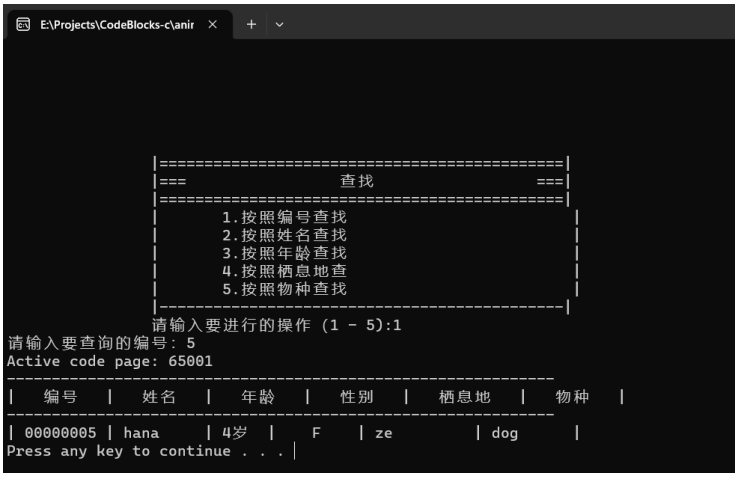


图 3.4: 查询记录

Chapter 4

测试

4.1 查询模块

测试用例 1: 按编号查询

输入:

```
struct Animal animals[] = { { 1, "Lion", "5", "Male", "Savannah", "Mammal" },
                            { 2, "Elephant", "10", "Female", "Jungle", "Mammal" },
                            { 3, "Tiger", "3", "Male", "Forest", "Mammal" } };
int animal_num = 3;
int target_num = 2;

SearchByNum(animals, animal_num, target_num);
```

预期输出:

```
-------------------------------------------------------------
|   编号   |   姓名   |   年龄   |   性别   |   栖息地   |   物种   |
-------------------------------------------------------------
| 00000002 | Elephant | 10岁  |    Female    | Jungle     | Mammal   |
```

实际输出:

```
-------------------------------------------------------------
|   编号   |   姓名   |   年龄   |   性别   |   栖息地   |   物种   |
-------------------------------------------------------------
| 00000002 | Elephant | 10岁  |    Female    | Jungle     | Mammal   |
```

4.2 修改模块

测试用例 1: 添加记录

输入:

```
record_num = 0;
```

```
Add();
```

预期输出：无输出，添加成功。
实际输出：无输出，添加成功。

测试用例 2: 删除记录

输入：

```
int target_nums[] = {2, 3};
Delete(target_nums);
```

预期输出：无输出，删除成功。
实际输出：无输出，删除成功。

测试用例 3: 修改记录

输入：

```
int target_nums[] = {1};
Change(target_nums);
```

预期输出：无输出，添加成功。
实际输出：无输出，添加成功。

测试用例 4: 删除记录

输入：

```
int target_nums[] = {2, 3};
Delete(target_nums);
```

预期输出：无输出，删除成功。
实际输出：无输出，删除成功。

测试用例 5: 修改记录

输入：

```
int target_nums[] = {1};
Change(target_nums);
```

预期输出：根据用户的选择，提示输入新的字段值。
实际输出：根据用户的选择，提示输入新的字段值。

Chapter 5

设计总结与展望

5.1 总结

校园动物管理系统是为了实现对校园内各类动物进行科学、规范、高效管理而设计的。通过本次实验，我们成功地完成了系统的设计和实现，并实现了基本的功能模块。

在实验过程中，我们首先进行了需求分析，明确了系统的功能需求。然后进行了总体设计，划分了系统的核心模块，并确定了各个模块的功能和相互关系。接着进行了详细设计与实现，编写了相应的代码，并进行了测试和调试，确保系统的正常运行。

通过实验，我们得出以下总结：

- 成功实现了校园动物管理系统的基本功能，包括动物信息的增删改查等操作，数据的存储，界面的友好性等。
- 使用 C 语言进行开发，具有良好的跨平台性和可移植性。
- 系统设计合理，模块划分清晰，代码结构清晰简洁，易于维护和扩展。
- 进一步了解了文件读写操作，数据结构的应用和界面设计等方面的知识。

虽然我们已经实现了基本功能，但仍有一些改进的空间：

- 完善系统的错误处理机制，对用户输入进行合法性检查，避免输入错误导致系统崩溃或数据损坏。
- 增加数据统计和分析模块，提供对动物数量、种类等数据的统计和分析功能，帮助管理员更好地了解校园动物的情况。
- 改进界面的交互设计，提供更多的操作选项和提示信息，提升用户体验和操作的便捷性。

校园动物管理系统是一个有潜力和发展前景的项目。未来可以考虑以下方面的拓展和改进：

- 引入图形界面：将系统的界面由命令行转为图形界面，提供更直观、友好的操作方式，提升用户体验。
- 引入更多功能模块：如动物医疗管理模块、动物喂养管理模块等，进一步完善系统的功能，提供更全面的动物管理服务。

- 引入智能化技术：如人工智能、物联网等技术，将智能化应用于校园动物管理系统中，提高管理的效率和准确性。例如，可以利用人工智能技术对动物行为进行分析和监测，提供预警和警报功能，帮助保护动物的安全和福利。

通过不断的创新和改进，校园动物管理系统将能够更好地满足学校管理动物的需求，推动校园内动物管理的现代化和智能化发展，为保护动物、促进人与动物的和谐共处做出贡献。

5.2 展望

校园动物管理系统是一个有潜力和发展前景的项目。未来可以考虑以下方面的拓展和改进：

- 引入图形界面：将系统的界面由命令行转为图形界面，提供更直观、友好的操作方式，提升用户体验。
- 引入更多功能模块：如动物医疗管理模块、动物喂养管理模块等，进一步完善系统的功能，提供更全面的动物管理服务。
- 引入智能化技术：如人工智能、物联网等技术，将智能化应用于校园动物管理系统中，提高管理的效率和准确性。例如，可以利用人工智能技术对动物行为进行分析和监测，提供预警和警报功能，帮助保护动物的安全和福利。

通过不断的创新和改进，校园动物管理系统将能够更好地满足学校管理动物的需求，推动校园内动物管理的现代化和智能化发展，为保护动物、促进人与动物的和谐共处做出贡献。

参考文献

[1] 许真珍, 蒋光远, 田琳琳. C 语言课程设计指导教程 [M]. 清华大学出版社, 2016.
Available at: `https://gitee.com/kim_tae_woo/animal-track.git`