Vocational High School

of Computer Programming and Innovation

Картина, която съдържа рисунка

Описанието е генерирано автоматично

Animal Shelter

Afra Yusrefova - [AFyusrefova@codingburgas.bg](mailto:AFyusrefova@codingburgas.bg)

Christina Bolashikova - [kvbolashikova18@codingburgas.bg](mailto:kvbolashikova18@codingburgas.bg)

Celine Feyzova - [safeyzova18@codingburgas.bg](mailto:safeyzova18@codingburgas.bg)

Reneta Topalska - [RBTopalska18@codingburgas.bg](mailto:RBTopalska18@codingburgas.bg)

1. Description of the project and goals

The project we have developed is a program designed to support the administrative activity of any animal shelter or (with small changes) any such animal business, for example a pet shop or a zoo. Its main objective is to facilitate the handling of data in the aforementioned economic units. We have made sure that the interface is as user-friendly as possible so that everyone can navigate through the program, and at every opportunity we have put guidance texts in the program that facilitate execution, giving instructions on what should or can be done, or notify when an error occurs.

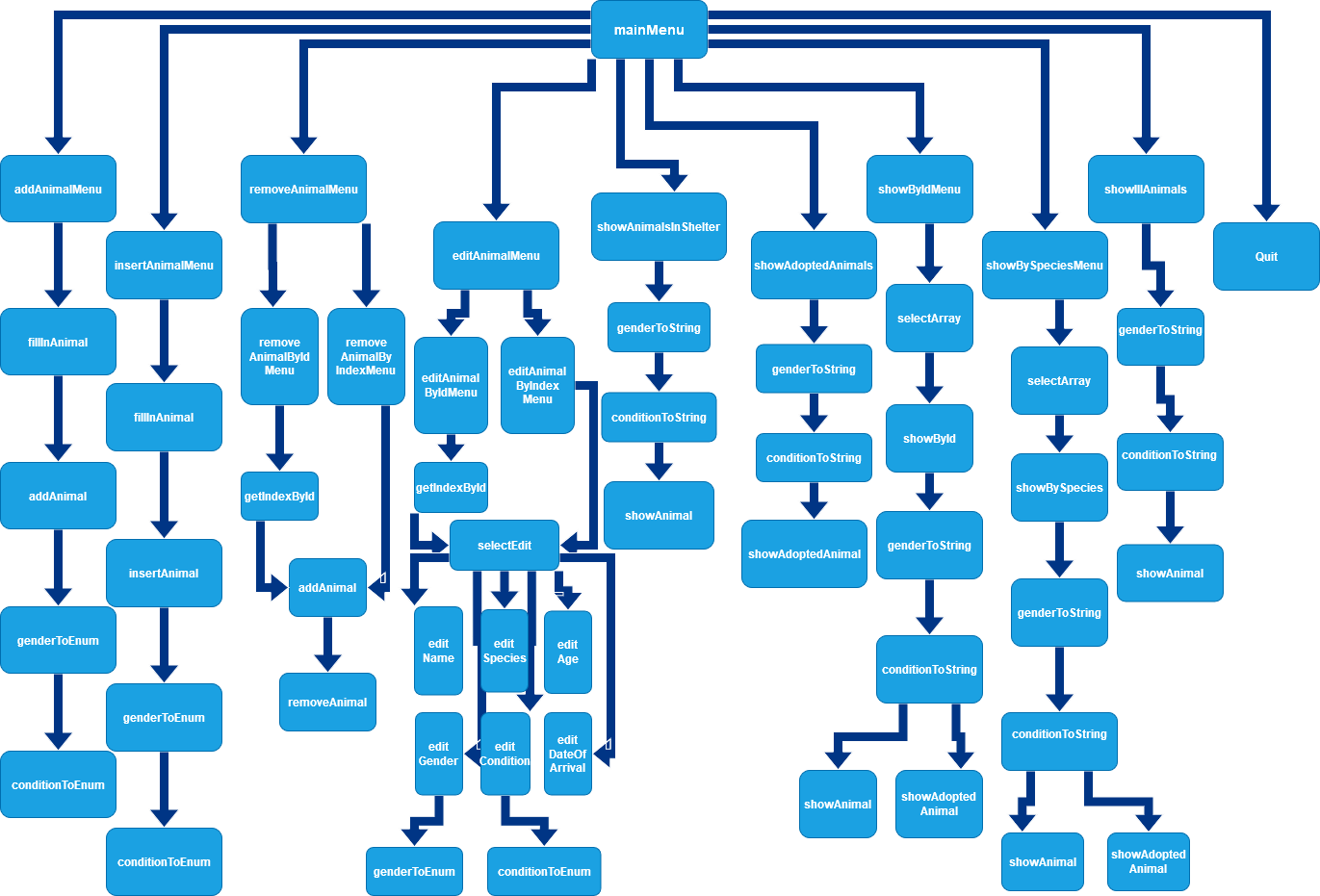
The main functionality is based on actions with two arrays of structures (of type ANIMAL) – one for the animals in the shelter (animals) and one for animals that have been adopted (adoptedAnimals), with both having predefined elements (for animals – 15, and for adoptedAnimals – 16). The ANIMAL structure consists of a total of 8 fields: **string name** – for the name; **int id** – for the ID number; **string species** – for the species of the animal; **float age** – for the age; enumeration type **gender gender** – for the gender; enumeration type **condition condition** – for the condition; **dateOfArrival,** which is a variable of the **DATE** structure – for the date of arrival at the shelter; **dateOfAdoption**, which is a variable of the **DATE** structure – for the date of the adoption. The **DATE** structure consists of three fields of type int – **day, month and year.**

The program begins with an introduction and greetings to the user. The main menu options are displayed below: adding an animal at the end of the animals array – **addAnimalMenu**; inserting an animal at a specific position in the animals array (according to the user-selected index) – **insertAnimalMenu**; removing an animal (according to the user-selected index or ID number) from the animals array and adding it to the array adoptedAnimals (with a new ID number) – **removeAnimalMenu**; editing animal data (animal is selected by its ID number or index by the user) – **editAnimalMenu**; showing all animals in the shelter (from the array animals) – **showAnimalsInShelter**; showing all animals that have been adopted (from the array adoptedAnimals) – **showAdoptedAnimals**; displaying a specific animal by the ID number entered (the user selects which one of the two arrays he/she wishes to browse) – **showByIdMenu**; showing all animals of a particular species (the user selects which one of the two arrays he/she wishes to browse) – **showBySpeciesMenu**; displaying all animals in bad condition[[1]](#footnote-1) (condition = BAD) - **showIllAnimals**; exiting the program.

1. Description of the functions used

|  |  |  |  |
| --- | --- | --- | --- |
| Name and type | Purpose | Arguments[[2]](#footnote-2) | Return value |
| void mainMenu | Displays main menu options and performs the corresponding function according to what the user has entered | ANIMAL \*animals, int &animalCount, ANIMAL \*adoptedAnimals, int &adoptedAnimalCount | - |
| void addAnimalMenu | Serves as the link between mainMenu and fillInAnimal and addAnimal | ANIMAL\* animals, int& animalCount | - |
| void insertAnimalMenu | Serves as a link between mainMenu and fillInAnimal and insertAnimal; the user enters at which position to insert the animal | ANIMAL\* animals, int& animalCount | - |
| void removeAnimalMenu | Checks whether the user wants to select the animal to be deleted by its ID number or by its index and calls the corresponding function – removeAnimalByIdMenu or removeAnimalByIndexMenu | ANIMAL \*animals, int &animalCount, ANIMAL \*adoptedAnimals, int &adoptedAnimalCount | - |
| void  editAnimalMenu | Checks whether the user wants to select the animal to be edited by its ID number or by its index and calls the corresponding function – editAnimalByIdMenu or editAnimalByIndexMenu | ANIMAL\* animals, int animalCount | - |
| void showAnimalsInShelter | Displays all animals that are part of the animals array | ANIMAL \*animals, int animalCount | - |
| void showAdoptedAnimals | Displays all animals that are part of the adoptedAnimals array | ANIMAL \*adoptedAnimals, int adoptedAnimalCount | - |
| void showByIdMenu | Asks the user which array he/she wants to work in, the ID number is entered and the **showById** function is executed accordingly with the parameters animals, animalCount, IdUser, selectedArray or adoptedAnimals, adoptedAnimalCount, IdUser, selectedArray | ANIMAL \*animals, int animalCount, ANIMAL \*adoptedAnimals, int adoptedanimalCount | - |
| void ShowBySpeciesMenu | Asks the user which array he/she wants to work in, the animal type is entered and the **showBySpecies** function is executed accordingly with the parameters animals, animalCount, speciesUser, selectedArray or adoptedAnimals, adoptedAnimalCount, speciesUser, selectedArray | ANIMAL \*animals, int animalCount, ANIMAL \*adoptedAnimals, int adoptedAnimalCount | - |
| void ShowIllAnimals | Checks if the animal is sick and if so, then it displays it on the console with **showAnimal** | ANIMAL \*animals, int animalCount | - |
| void fillInAnimal | Serves to enter animal data | ANIMAL& newAnimal, string& gender, string& condition | - |
| void genderToEnum / conditionToEnum | Converts the gender/condition of the animal from string into enumeration type gender/condition | ANIMAL \*animals, string gender / condition, int i | - |
| int getIndexById | Finds the index of the animal that matches the ID entered by the user | ANIMAL\* animals, int animalCount,int idUser | The index variable |
| string genderToString / conditionToString | Converts the gender/condition of the animal from enumeration type gender/condition into string | ANIMAL \*animals,int i | The gender / condition variable |
| void showAnimal | Displays an animal’s profile on the console | ANIMAL \*animals,int i,string gender,string condition | - |

1. Diagram describing the menus / functions in the program



1. Animals in poor condition can only be present in the *animals* array, since in order to transfer an animal to the *adoptedAnimals* array, the animal has to be in good condition (condition = GOOD), and in addition all present animals in *adoptedAnimals* are in good condition. [↑](#footnote-ref-1)
2. ANIMAL \*animals – pointer to the *animals* array; int &animalCount – alias of the variable *animalCount*, which shows the number of elements in the *animals* array and the first free space in the array; ANIMAL \*adoptedAnimals – pointer to the array *adoptedAnimals*; int &adoptedAnimalsCount – alias of the *adoptedAnimalsCount* variable, which shows the number of elements in the array *adoptedAnimals* and the first free space in the array. [↑](#footnote-ref-2)