Linked Lists History Project

Table of Contents

1. **Authors**…………………………………………………………………………………………………………………………………………...**3**
2. **Programs we used**…………………………………………………………………………………………………………………………..**3**
3. **Summary**…………………………………………………………………………………………………………………………………………**4**

3.1. Goals………………………………………………………………………………………………………………………………………….4

3.2. Stages of realization……………………………………………………………………………………………………………………4

3.3. Flowchart…………………………………………………………………………………………………………………………………..5

3.4. Functions we used………………………………………………………………………………………………………………………6

**4. Conclusion**…………………………………………………………………………………………………………………………………………10

1. Authors

* Velimir Vladimirov Dimitrov – 10B class
  + Scrum Trainer
* Mariya Dimitrova Kostova – 10V class
  + Backend Developer
* Teodora Nikolaeva Kompanska – 10A class
  + Frontend Developer
* Danislav Antoniev Todorov – 10G class
  + QA

1. Programs we used

* Visual Studio 2019 – to make our program and our QA
* Excel – for our QA documentation
* Word – for our project documentation
* PowerPoint – for our project presentation
* Github – for better teamwork
* Discord and Teams – for communication

1. Summary

## 3.1. Goals

Our goal is to make a program where you can learn about the Bulgarian monarchs from different eras. There are also some functions that you can use such as inserting, editing, searching, deleting, sorting and ordering monarchs. The program also has a function where you can enter a year and view which monarch had ruled then and includes quiz for this monarchs, asking from which to which year had they ruled.

## 3.2. Stages of realization

***First stage (planning)*** – Firstly, we have to make groups to communicate. After that we distributed the roles and considered our idea of the program and then our scrum trainer made repository in Github.

***Second stage (realization)*** – When we started to work, we discussed every day how to make the functions given by the regulation and what new functions we can put in the program. We also improved our idea by making a quiz.

***Third stage (presenting our project)*** – After we finished our work on code, we have to make documentation and presentation to present our project easier.

Diagram

Description automatically generated

## 3.3. Flow chart

## 3.4. Functions we used

|  |  |  |
| --- | --- | --- |
| **Function name** | **Type of the function** | **Purpose** |
| menuRoof() | void | Draws part of the main menu |
| menuTop() | void | Draws part of the main menu |
| menuMiniRoof() | void | Draws part of the main menu |
| menuBottom() | void | Draws part of the main menu |
| menuCastle() | void | Draws part of the main menu |
| menuTitle() | void | Draws part of the main menu |
| menuPort() | void | Draws part of the main menu |
| menuOptionsBorder() | void | Draws part of the main menu |
| menuOptions(int counter) | void | Draws part of the main menu |
| menu(int counter) | void | Draws the main menu |
| registrationFormLeftBorder() | void | Draws the left, up and down registration form's border |
| registrationFormRightBorder() | void | Draws registration form's right border |
| registrationFormOptions() | void | Displays registration form's options |
| registrationFormArrow(int counter) | void | Moves the arrow in the register form |
| useInformation() | void | Draws ‘How to use’ menu |
| exit() | void | Draws label ‘Goodbye’ |
| choosingMenuCrown() | void | Draws part of the choosing menu |
| choosingMenuSword() | void | Draws part of the choosing menu |
| choosingMenuSwordTwo() | void | Draws part of the choosing menu |
| choosingMenu() | void | Draws part of the choosing menu |
| choosingMenuOptions(int counter) | void | Draws part of the choosing menu |
| choosingMenu(int counter) | void | Draws the choosing menu |
| title() | void | Draws the ‘BG MONARCHS’ title |
| leftBorder() | void | Draws left border |
| rightBorder() | void | Draws right border |
| askInput() | void | Draws options |
| arrow(int counter) | void | Moves the arrow in the menus |

|  |  |  |
| --- | --- | --- |
| hintFirstBgEmpire(int number, int id, int correct, int incorrect, int skipped) | void | Function for giving hint if the user doesn't know what is the start reign year |
| hintSecondBgEmpire(int number, int id, int correct, int incorrect, int skipped) | void | Function for giving hint if the user doesn't know what is the start reign year |
| viewInfoMenu() | void | Function for viewing the options when user chooses to view monarch’s info |
| insertMenu() | void | Function for viewing the options when user chooses to insert a monarch |
| searchMenu() | void | Function for viewing the options when user chooses to search a monarch |
| orderMenu() | void | Function for viewing the options when user chooses to order the monarch and the start reign year |
| sortMenu() | void | Function for viewing the options when user chooses to sort the monarchs |
| quizMenu() | void | Function for viewing the options when user chooses to start the quiz |
| contentFirstBulgarianEmpire(int id) | void | Function for linked list’s content – First Bulgarian Empire |
| contentSecondBulgarianEmpire(int id) | void | Function for linked list’s content – Second Bulgarian Empire |
| contentAfterLiberation(int id) | void | Function for linked list’s content – After Liberation of Bulgaria |
| printYearFirst(Node\* head) | void | Prints linked list’s elements so that first thing is the start reign year |
| printNameFirst(Node\* head) | void | Prints linked list’s elements so that first thing is the monarch’s name |
| viewInformationByYear(Node\* head) | void | Prints information about given monarch when user chooses the start reign year |
| viewInformationByName(Node\* head) | void | Print information about given monarch when user chooses his name |
| beginInsert(Node\*\* head, int startReignYear, int endReignYear, string name, string information) | void | Inserts Node in the beginning of the linked list |
| endInsert(Node\*\* head, int startReignYear, int endReignYear, string name, string information) | void | Inserts Node in the end of the linked list |

|  |  |  |
| --- | --- | --- |
| insertNodeAfterYear(Node\* head, int startReignYear) | void | Searches what year has the user input |
| insertAfterNode(Node\* temp, Node\* head, int startReignYear, int endReignYear, string name, string information) | void | Inserts Node after given Year |
| findNodeByYear(Node\* head, int startReignYear, int id) | void | Finds Node by given start reign year |
| findNodeByName(Node\* head, string name, int id) | void | Finds Node by given monarch’s name |
| deleteNode(Node\*\* head, int startReignYear) | void | Deletes Node after given start reign year |
| editNode(Node\* head, int startReignYear) | void | Edits Node by searching it by the start reign year |
| sortDescending(Node\* head) | void | Sorts the linked list in descending way |
| sortAscending(Node\* head) | void | Sorts the linked list in ascending way |
| viewMonarch(Node\* head, int id) | void | Asks the user to input year and then searches which monarch had ruled then |
| grayToBin(int num) | int | Converts from gray to binary number |
| choosingInsertBegin(Node\* Head) | void | Asks the user to input variables with which the beginInsert function will work |
| void choosingInsertEnd(Node\* Head) | void | Asks the user to input variables with which the endInsert will work |
| void choosingInsertAfterNode(Node\* temp, Node\* head) | void | Asks the user to input variables with which the insertAfterNode will work |
| void choosingYearSearch(Node\* Head, int id) | void | Asks the user to input variables with which findNodeByYear will work |
| void choosingNameSearch(Node\* Head, int id) | void | Asks the user to input variables with which the findNodeByName will work |

|  |  |  |
| --- | --- | --- |
| void choosingEditElement(Node\* Head) | void | Asks the user to input variables with which the editNode will work |
| menuInput() | bool | Goes around the main menu |
| choosingMenuInput(int id) | bool | Goes around the choosing menu |
| contentMenuInputYearFirst(Node\* Head) | bool | Goes around the monarchs content menu when the start reign year is shown first |
| contentMenuInputYearFirstGuest(Node\* Head) | bool | Goes around the monarchs content menu when the start reign year is shown first and the user is guest |
| contentMenuInputNameFirst(Node\* Head) | bool | Goes around the monarchs content menu when the monarch’s name is shown first |
| contentMenuInputNameFirstGuest(Node\* Head) | bool | Goes around the monarchs content menu when the monarch’s name is shown first and the user is guest |
| insertChoice(Node\* Head) | bool | Goes around the insert monarch menu |
| searchChoice(Node\* Head, int id) | bool | Goes around the search monarch menu |
| orderChoice(Node\* Head, int id) | bool | Goes around the order monarchs menu |
| sortChoice(Node\* Head, int id) | bool | Goes around the sort monarchs menu |
| viewInfoChoice(Node\* Head, int id) | bool | Goes around the view information about monarch menu |
| quizChoice() | bool | Goes around the quiz menu |
| settingsInput(Node\* Head) | void | Prints settings and asks user to choose an option |
| startQuizFirstBgEmpire(int id, int correct, int incorrect, int skipped) | void | Starts the quiz about First Bulgarian Empire |
| startQuizSecondBgEmpire(int id, int correct, int incorrect, int skipped) | void | Starts the quiz about Second Bulgarian Empire |
| login() | void | Log in function |
| registration() | void | Register function |
| forgottenPassword() | void | Function to view password if you have forgot it |

1. Conclusion

We have made all the functions given by the regulation and also have added more. Our program has quiz that is very interesting and something new can be learn from it.

To improve our project we can add more functions, more monarchs’ content and more information about them. Also we can add new questions in the quiz with given information about any monarch and the user will have to guess who it is.