Seat assignment program

Team:

- Alexander Zink
- Jakob Reinelt
- Laszlo Magocsi
- Markus Genner

Introduction

The appended program is used to create a seating plan in compliance with current social distancing regulations.

The seating plan is created according to the user input. Seats can be assigned to a student using the students ID. All assignments and the seating plan itself are saved to a logfile that is saved on a predetermined path.

classroom.h

This headerfile contains all functions that are in relation to creating the seating plan.

It starts of by creating the seat struct. This struct has two fields:

*student: The student's ID.

*nextSeat: A pointer to the next seat in the classroom.

With the help of this struct, another struct is created. The classroom struct represents the seating plan. It too has two fields.

*firstSeat: A pointer to the first seat in the classroom.

*lastSeat: A pointer to the last seat in the classroom.

The classroomCreate() function creates a new classroom when called and returns a pointer to it. It allocates memory for a new classroom struct and initializes its fields to NULL.

The classroomAppendLastSeat() function appends a new student to the last empty seat in the classroom. It allocates memory for a new seat struct, initializes its fields, and then appends it to the end of the classroom.

The classroomSearchStudAddr() function searches for a student in the classroom and returns a pointer to their seat. It searches for a student in the classroom by using their ID. It starts at the first seat in the classroom and compares the student ID of each seat to the ID that was passed in. If the IDs match, the function returns a pointer to the seat. If the ID does not match, the function continues searching until it reaches the end of the classroom. If the ID is not found, the function returns NULL.

The classroomSearchStudOrd() function searches for a student in the classroom and returns their seat number. It searches for a student in the classroom by their seat number. It starts at the first seat in the classroom and compares the seat number of each seat to the seat number that was passed in. If the seat numbers match, the function returns the index of the seat. If the seat numbers do not match, the function continues searching until it reaches the end of the classroom. If the seat number is not found, the function returns -1.

The classroomSearchOrdAddr() function searches for a seat in the classroom and returns a pointer to it. It is used to searches for a seat in the classroom by its index. It starts at the first seat in the classroom and compares the index of each seat to the index that was passed in. If the indexes match, the function returns a pointer to the seat. If the indexes do not match, the function continues searching until it reaches the end of the classroom. If the index is not found, the function returns NULL.

The classroomSearchOrdStud() function searches for a seat in the classroom and returns the student ID of the student sitting in it. It searches for a seat in the classroom by its index and returns the student ID of the student sitting in it. It starts at the first seat in the classroom and compares the index of each seat to the index that was passed in. If the indexes match, the function returns the student ID of the student sitting in the seat. If the indexes do not match, the function continues searching until it reaches the end of the classroom. If the index is not found, the function returns NULL.

The classroomPrintWhole() function prints a list of all the students in the classroom to the console and to a previously defined logfile. It starts at the first seat in the classroom and prints the student ID of the student sitting in each seat. It continues printing until it reaches the end of the classroom. It prints to the console as well as to a predefined logfile.

The classroomPrintPartial() function prints a list of the students in a specified neighbourhood of a given seat. It prints a list of the students in a specified neighborhood of a given seat. It starts at the given seat and prints the student IDs of the students sitting in the seats directly to the left, right, above, and below it. It then repeats this process for each of the students in the neighborhood, until it reaches the edge of the classroom.