

**Kурсов проект по ООП**

**Билети**

Преподавател:

Доц.д-р. Калин Георгиев

Изготвил: Яница Хаджиева ,

2курс ,

фн.71968

**1**.Условие

Проект 14: Билети

Да се напише програма, реализираща информационна система, която обслужва билетна каса. Програмата съхранява и обработва необходимите данни във файл.

Представленията се играят в няколко зали, всяка от които има номер, брой редове и брой места на всеки ред. Залите са предварително зададени.

След като приложението отвори даден файл, то трябва да може да извършва посочените по-долу операции, в допълнение на общите операции (open, close, save, save as, help и exit):

|  |  |
| --- | --- |
| addevent <date> <hall> <name> | Добавя ново представление на дата <date> с име <name> в зала <hall>. Ако в тази зала вече има друго представление на същата дата, командата да върне грешка. |
| freeseats <date> <name> | Извежда справка за свободните места за представление с име <name> на дата <date>  (непродадени и незапазени билети). |
| book <row> <seat> <date> <name> <note> | Запазва билет за представление с име <name> на <date> на ред <row> и място <seat>, като добавя бележка <note>. |
| unbook <row> <seat> <date> <name> | Отменя резервация за представление с име <name> на <date> на ред <row> и място <seat>. |
| buy <row> <seat> <date> <name> | Закупува билет за представление с име <name> на <date> на ред <row> и място <seat>. За всеки билет се издава уникален сложен код, който съдържа информация за съответното място |
| bookings [<date>] [<name>] | Извежда справка за запазените, но неплатени (незакупени) билети за  представление с име <name> на <date>. Ако <name> е пропуснато, извежда информация за всички представления на дадената дата. Ако <date> е пропуснато, извежда информация за всички дати. |
| check <code> | Прави проверка за валидност на билет, като по дадения код <code> се извлича номера на мястото или се връща грешка, ако кодът е невалиден). |
| report <from> <to> [<hall>] | Извежда справка за закупени билети от дата <from> до дата <to> в зала <hall>, като извежда всички изнесени представления в залата и за всяко отделно представление се извежда и количеството продадени билети. Ако <hall> е пропуснато, извежда информация за всички зали. |

1. Схема на проекта:



**3**.Операции , които поддържа приложението:

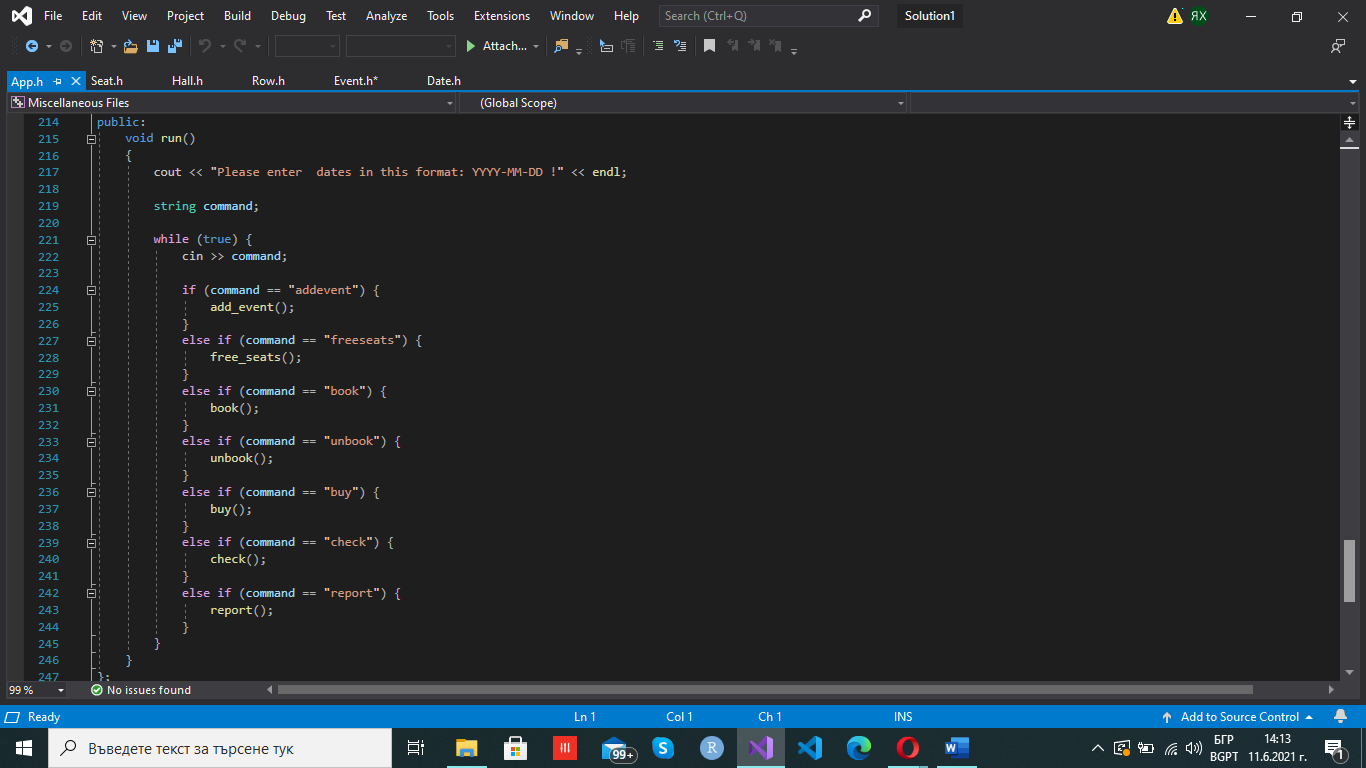
|  |
| --- |
| * addevent |
| * freeseats |
| * book |
| * unbook |
| * buy |
| * bookings |
| * check |
| * report <from> <to> |

**4**.Архитектура на проекта

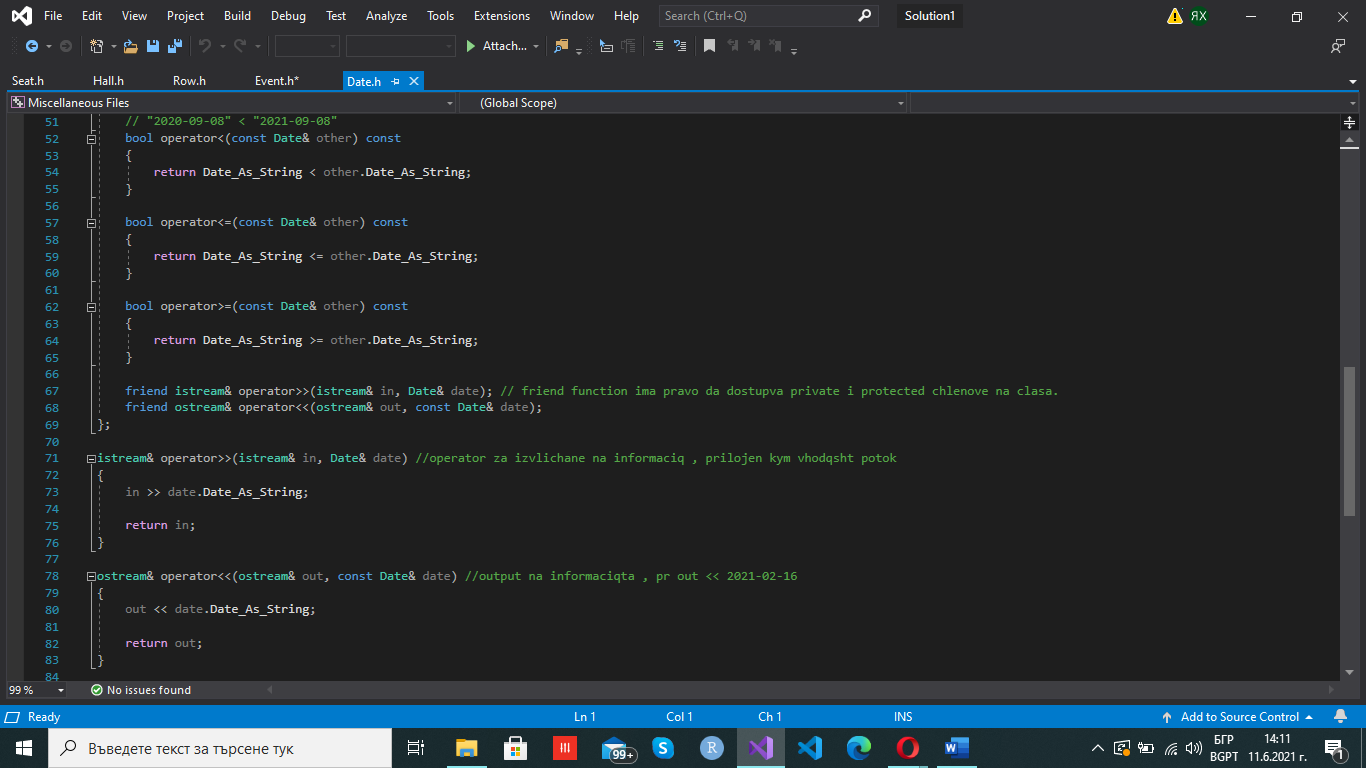
Програмата се състои от няколко класа :

**Seat, Hall, Row, Event, Date, App**;

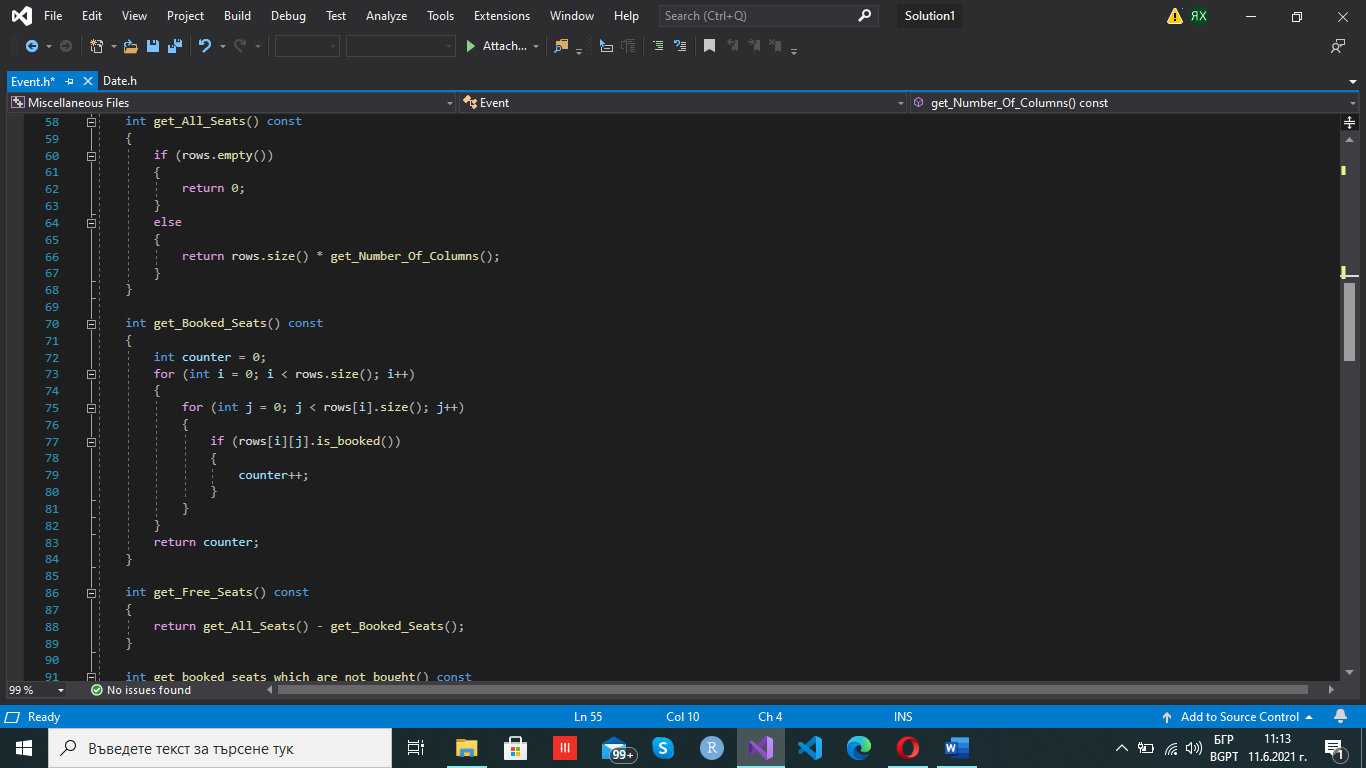
**App** е класът , който съдържа цялата функционалността на програмата. Всяка потребителска команда има съответна функция в този клас. В него също е и основната функция Run().

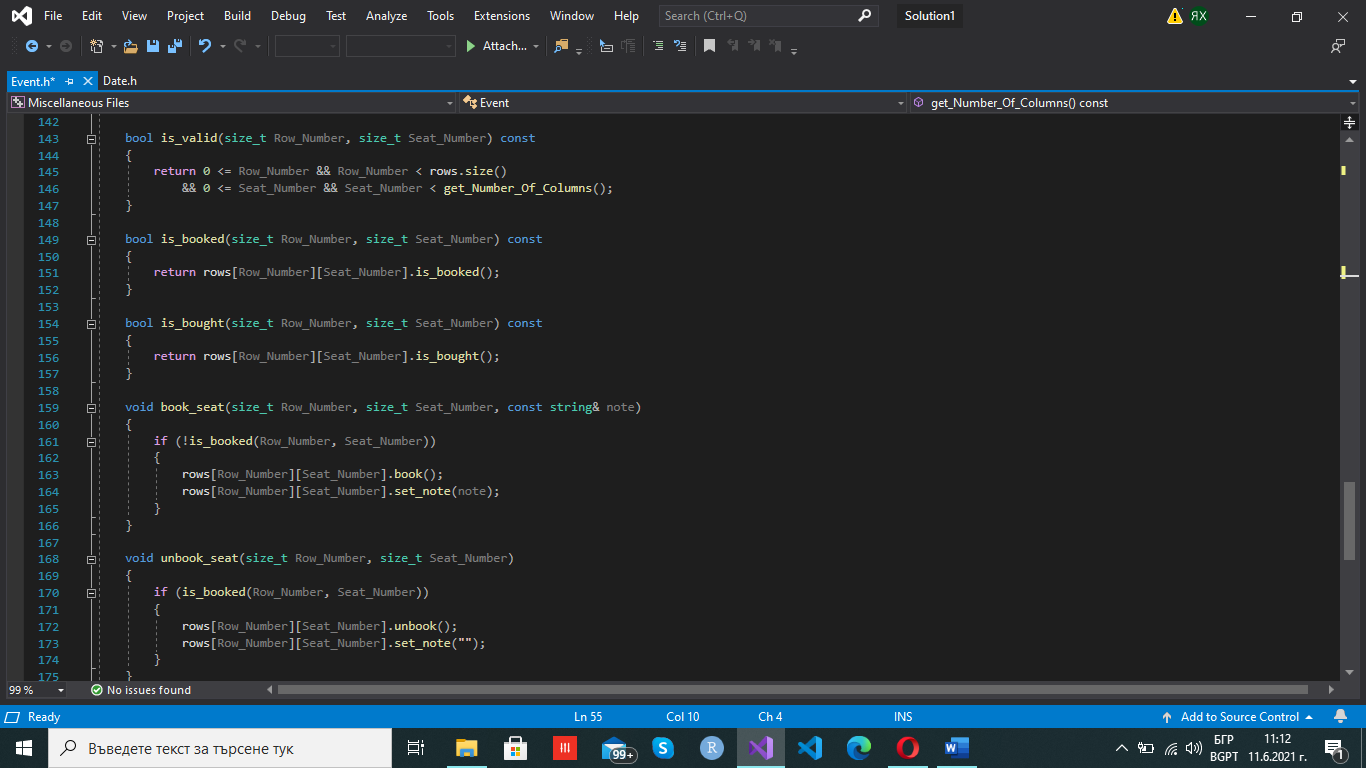


**Date** е класът , който преобразува датите във формат стринг , и работи с операторите „== , < , <= , => , << , >> “.



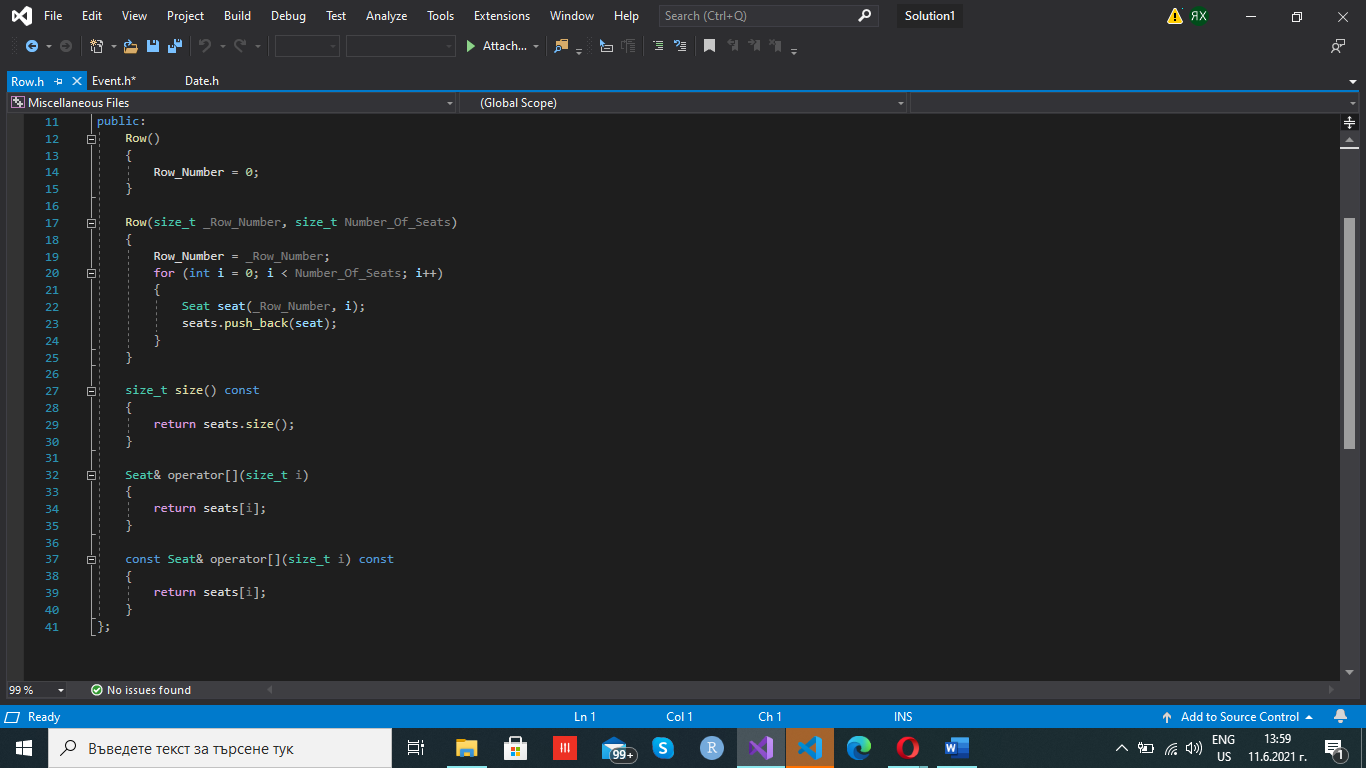
**Event** е класът, който работи с гетърите на основните променливи:



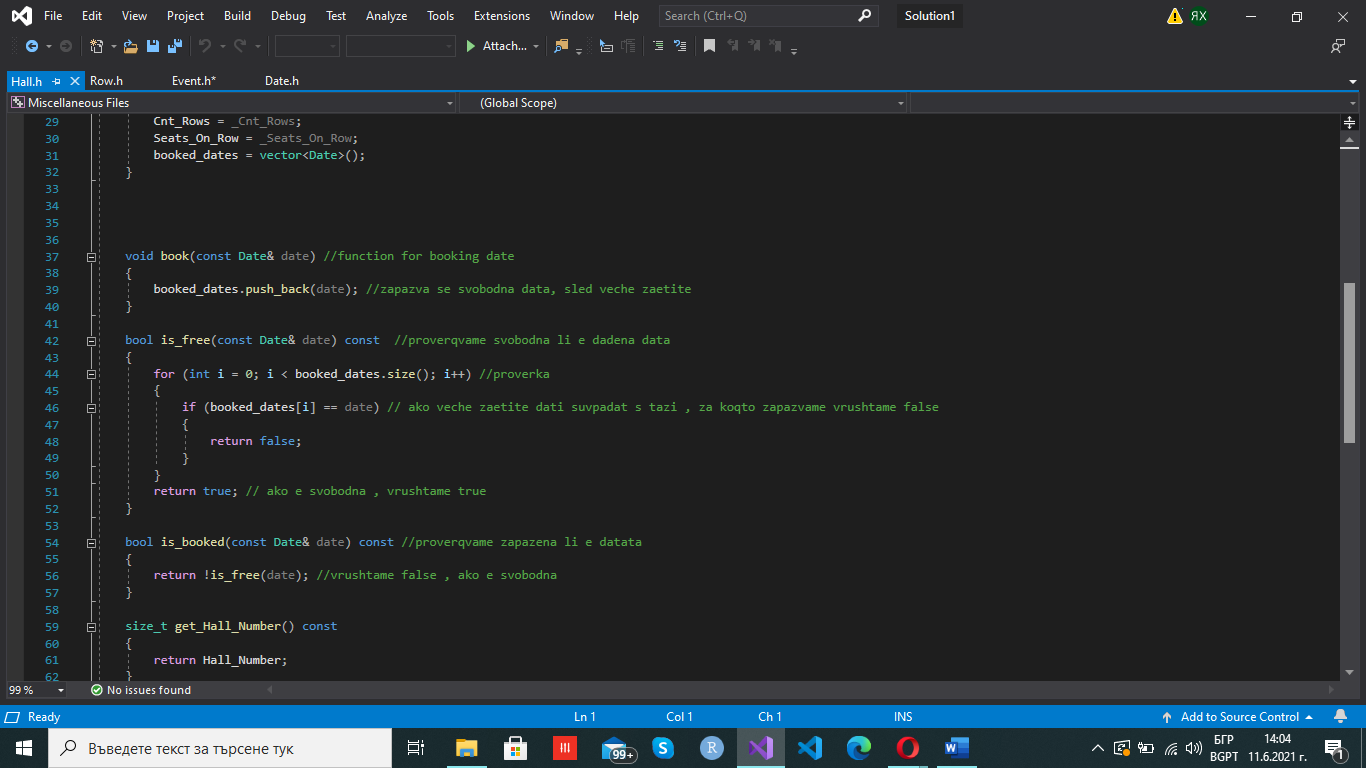
И също така в него има функции , за проверки: 

**Row** е класът , в който има функция size() , която взима всички елементи , които се съдържат във вектора seats.

Класа поддържа също и оператор [].



**Hall** съдържа функция за записване на свободна дата и такава , която проверява дали дадена дата е свободна , а ако не е извежда съобщение за грешка.



**Източници:**

<https://www.tutorialspoint.com/cplusplus/cpp_date_time.htm>

/https://www.cplusplus.com/reference/istream/istream/peek/

**Seat** е класът , в който има функция , която генерира уникален сериен номер , за билет , който е закупен и функция за купуване на билета.

