

Итоговый проект «Разработка корпоративного MVC- приложения железнодорожной компании»

По ДПП «Введение в разработку корпоративных приложений на Java»

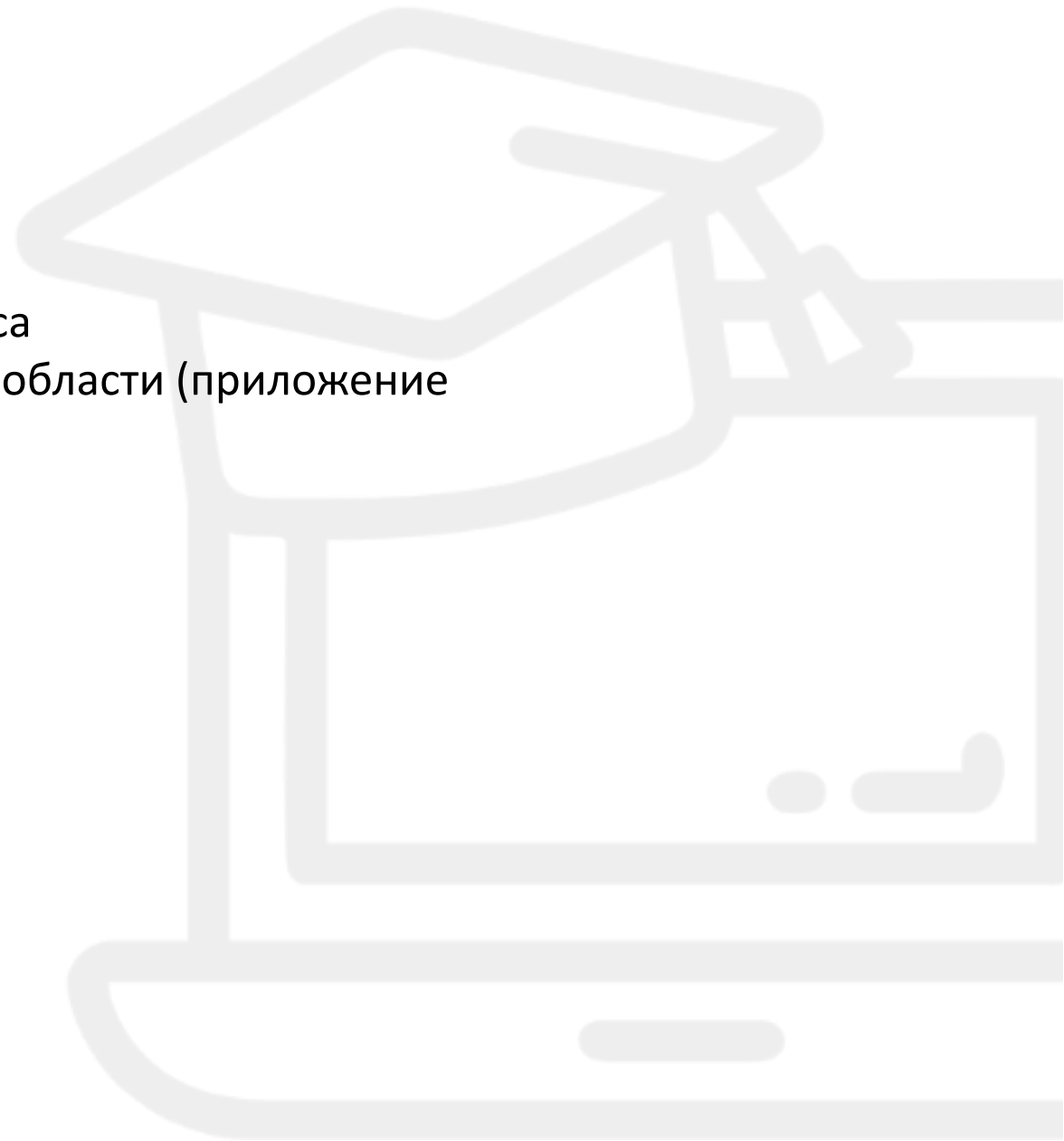
Выполнили:

студент гр.4932 Иванов Никита

студент гр.4931 Александров Павел

ЦЕЛЬ РАБОТЫ

- Обобщить навыки, полученные в ходе изучения курса
- Реализовать прототип приложения для предметной области (приложение железнодорожной компании)



ОСНОВНЫЕ ЗАДАЧИ

Серверная часть (Иванов Никита):

- Поиск поездов по параметрам
- Просмотр доступных и купленных мест
- Покупка доступных мест
- REST API для интеграции сторонних сервисов

Клиентская часть (Александров Павел):

- Страница с поиском поездов
- Страница с доступными и недоступными местами
- Механизм покупки мест




ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

- Spring boot
- Spring doc (open api)
- Spring data
- PostgreSQL
- Spring MVC
- Thymeleaf
- Bootstrap 5



Примеры интерфейса разработанной программы. Поиск поездов.

 Trains API


Departure City

Arrival City

Departure Time
05.06.2023, 00:00

Arrival Time
07.06.2023, 00:00

Search

 TRN011

06-06-2023 17:54
Cairo


06-06-2023 14:07
Kyoto

Compartment 12/26 from 105,45 P

Reserved seat 7/18 from 1 479,97 P

Buy

Buy

 TRN015

06-06-2023 23:56
Reykjavik


06-06-2023 15:36
Warsaw

Compartment 13/22 from 113,81 P

Reserved seat 10/22 from 1 003,76 P

Buy

Buy

 TRNE007

05-06-2023 17:17
Sofia


05-06-2023 19:51
San Francisco

Compartment 7/22 from 112,09 P

Reserved seat 12/30 from 1 013,96 P

Buy

Buy

 © 2023

Departure Time
ДД.ММ.ГГГГ, --:--


Arrival Time
ДД.ММ.ГГГГ, --:--

май 2023 г.


пн	вт	ср	чт	пт	сб	вс
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Очистить

Примеры интерфейса разработанной программы. Доступные места в поезде с сортировкой по номеру места.

 Trains API

Train

 TRNE013
29-05-2023 05:44
Kyoto

→

29-05-2023 12:47
Helsinki

Compartment 11/22 from 101,86 P [Buy](#)

Reserved seat 10/20 from 1 076,62 P [Buy](#)

Available Reserved Seat places

[Sort by number](#) [Sort by price](#)

≡ 3
3 548,09 P

≡ 5
4 178,30 P

≡ 12
3 240,57 P

≡ 14
3 340,79 P

≡ 22
4 394,81 P

≡ 26
4 474,48 P


≡ 28
1 076,62 P

≡ 32
2 332,28 P


≡ 40
1 681,14 P

≡ 41
3 945,21 P

Примеры интерфейса разработанной программы. Доступные места в поезде с сортировкой по цене места.

 Trains API

Train

 TRNE013
29-05-2023 05:44
Kyoto

→


29-05-2023 12:47
Helsinki


Compartment 11/22 from 101,86 P [Buy](#)


Reserved seat 10/20 from 1 076,62 P [Buy](#)


Available Reserved Seat places


[Sort by number](#) [Sort by price](#)


 28
1 076,62 P


 40
1 681,14 P


 32
2 332,28 P


 12
3 240,57 P


 14
3 340,79 P

 3
3 548,09 P

 41
3 945,21 P

 5
4 178,30 P

 22
4 394,81 P

 26
4 474,48 P

Примеры интерфейса разработанной программы. Недоступные места.

Unavailable Reserved Seat places

≡ 1

4 205,94 P

≡ 6

2 551,48 P

≡ 11

1 242,16 P

≡ 12

3 240,57 P

≡ 13

4 972,62 P

≡ 20

4 695,68 P

≡ 23

1 736,91 P

≡ 30

1 359,60 P

≡ 35

2 206,71 P

≡ 37

4 936,04 P

≡ 42

3 444,07 P





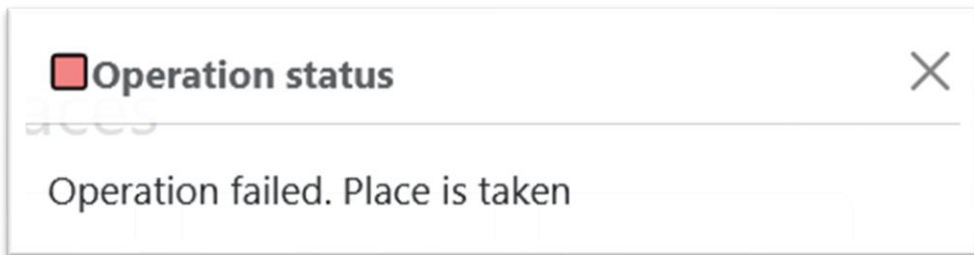
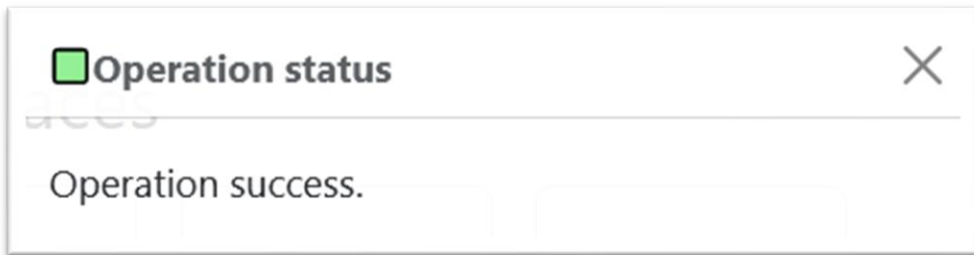
Примеры интерфейса разработанной программы. Диалоговое окно для покупки места.

guap.ru

A screenshot of a web application interface. In the foreground, a white dialog box titled "Buy place #12" is centered. It contains three input fields: "First Name", "Second Name", and a dropdown menu currently showing "Male". At the bottom right of the dialog are two buttons: a grey "Close" button and a green "Buy" button. The background is a blurred view of a seat selection interface. It shows a grid of seats with icons (like a bus icon) and text such as "TRNE013", "29-05-2023", "Kyoto", and "Avail". Below the grid, there are five columns of seat options, each with a bus icon, a number, and a price: "40 1 681,14 P", "32 2 332,28 P", "12 3 240,57 P", "14 3 340,79 P", and "3 3 548,09 P".

Примеры интерфейса разработанной программы. Уведомления.

- При успешной покупке места выводится уведомление о успешной покупке
- При неудачной покупке места (место уже занято) выводится уведомление об ошибке



Примеры интерфейса разработанной программы. REST API.

train-controller

GET /api/train/{id} Get element by id

PUT /api/train/{id} Edit element by id

DELETE /api/train/{id} Delete element by id

POST /api/train/search Search element by example

GET /api/train/ Get a list of all elements

POST /api/train/ Add element

train-express-controller

GET /api/train-express/{id} Get element by id

PUT /api/train-express/{id} Edit element by id

DELETE /api/train-express/{id} Delete element by id

POST /api/train-express/search Search element by example

GET /api/train-express/ Get a list of all elements

POST /api/train-express/ Add element

Примеры интерфейса разработанной программы. REST API.

place-controller

GET /api/place/{id} Get element by id

PUT /api/place/{id} Edit element by id

DELETE /api/place/{id} Delete element by id

POST /api/place/search Search element by example

POST /api/place/insert/train/{id}

POST /api/place/insert/train-express/{id}

GET /api/place/ Get a list of all elements

DELETE /api/place/{id}/passenger Remove passenger from place by place id

passenger-controller

GET /api/passenger/{id} Get element by id

PUT /api/passenger/{id} Edit element by id

DELETE /api/passenger/{id} Delete element by id

POST /api/passenger/{passengerID}/place/{placeID} Link existing element with the parent

POST /api/passenger/search Search element by example

POST /api/passenger/insert/place/{id} Insert a new element and link with the parent

GET /api/passenger/ Get a list of all elements

POST /api/passenger/ Add element

Примеры интерфейса разработанной программы. REST API.

```
Train ∨ {
  id
  number
  arrivalCity
  departureCity
  arrivalTime
  departureTime
  places

  integer($int64)
  string
  string
  string
  string
  string

  ∨ [Place ∨ {
    id
    number
    placeClass

    integer($int64)
    integer($int32)
    string
    Enum:

    ∨ [ Reserved Seat, Compartment ]

    passenger

    Passenger ∨ {
      id
      firstName
      secondName
      gender

      integer($int64)
      string
      string
      string
      Enum:

      ∨ [ Male, Female ]

    }

    price
    fullName
    available
    priceStr

    }
  ]
  string

  className
}
```

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ РАЗРАБОТАННОЙ ПРОГРАММЫ

- Поиск поездов, просмотр свободных мест, покупка мест
- REST API: CRUD методы для поездов, мест и пассажиров

Спасибо за внимание

Выполнил:

студент гр.4932 Иванов Никита

студент гр.4931 Александров Павел