

Комитет по образованию Правительства Санкт-Петербурга
**САНКТ-ПЕТЕРБУРГСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Отчет по практической работе
МДК 01.02 «Разработка мобильных приложений»
Разработка интерактивного приложения карт
с использованием REST API

Выполнил
студент группы 493:
Лукьянов И. А.
Преподаватель: Фомин А.В.

Санкт-Петербург 2022

Структура базы данных

База данных состоит из 5 таблиц:

1. Tile – хранит загруженные тайлы.
2. Layer – хранит настройки дополнительных слоев.
3. CacheTime – хранит время кэширования.
4. LastPosition – хранит последнее место на карте.
5. APIEndPoint – хранит конечную точку API.

ER диаграмма представлена на рисунке 1.

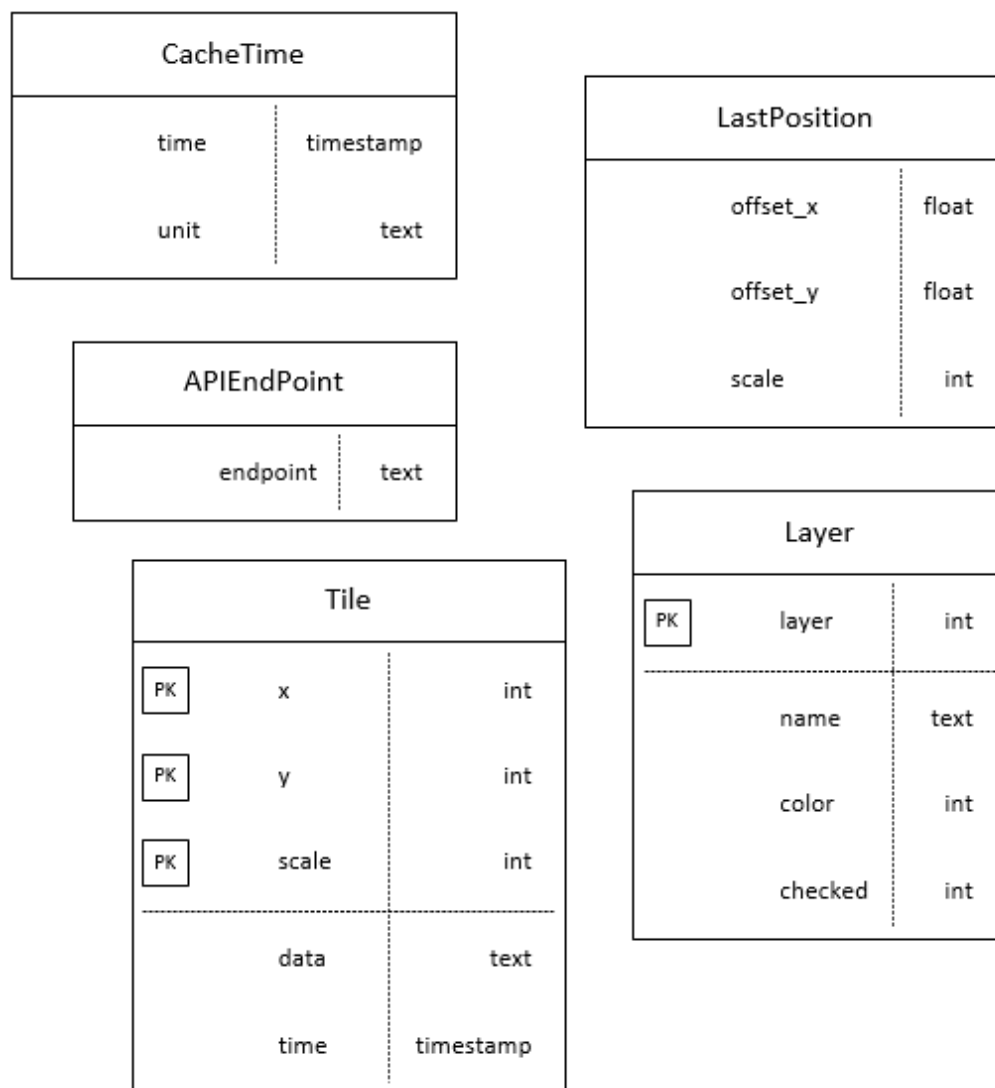


Рисунок 1 – ER диаграмма базы данных

Таблица Tile

Содержит сведения о загруженных тайлах. Таблица состоит из пяти столбцов:

1. x – координата тайла по горизонтали.
2. y – координата тайла по вертикали.
3. scale – уровень масштабирования.
4. data – изображение в формате строки.
5. time – время создания тайла.

Подробное описание столбцов представлено на рисунке 2.

Tile		loaded tiles						
#	name	type	size	default	primary	foreign	unique	description
1	x	int	-		yes	-	yes	x coordinate
2	y	int	-		yes	-	yes	y coordinate
3	scale	int	-		yes	-	yes	scale level
4	data	text	-		no	-	no	image in base64 type
5	time	timestamp	-		no	-	no	creation time

Рисунок 2 – Описание столбцов таблицы Tile

Таблица Layer

Содержит сведения о настройках дополнительных слоев. Таблица состоит из четырех столбцов:

1. layer – уникальный идентификатор слоя.
2. name – название слоя.
3. color – цвет слоя в численном выражении.
4. checked – статус слоя: выбран или нет.

Подробное описание столбцов представлено на рисунке 3.

Layer		layer settings						
#	name	type	size	default	primary	foreign	unique	description
1	layer	int	-		yes	-	yes	unique identifier
2	name	text	-		no	-	no	layer name
3	color	int	-		no	-	no	layer color
4	checked	int	-		no	-	no	layer status

Рисунок 3 – Описание столбцов таблицы Layer

Таблица LastPosition

Содержит сведения о последнем месте на карте. Таблица состоит из трёх столбцов:

1. offset_x – координата по горизонтали.

2. offset_y – координата по вертикали.

3. scale – уровень масштабирования.

Подробное описание столбцов представлено на рисунке 4.

<i>LastPosition</i>		<i>last fixed position</i>						
#	name	type	size	default	primary	foreign	unique	description
1	offset_x	float	-		no	-	no	offset x coordinate
2	offset_y	float	-		no	-	no	offset y coordinate
3	scale	int	-		no	-	no	scale level

Рисунок 4 – Описание столбцов таблицы LastPosition

Таблица CacheTime

Содержит сведения о времени кэширования тайлов. Таблица состоит из двух столбцов:

1. time – время кэширования.

2. unit – единица измерения: секунды/минуты/часы/дни.

Подробное описание столбцов представлено на рисунке 5.

<i>CacheTime</i>		<i>time of caching</i>						
#	name	type	size	default	primary	foreign	unique	description
1	time	timestamp	-		no	-	no	time of caching
2	unit	int	-		no	-	no	unit of measurement

Рисунок 5 – Описание столбцов таблицы CacheTime

Таблица APIEndPoint

Содержит сведения о конечной точке API. Таблица состоит из одного столбца:

1. endpoint – конечная точка API.

Подробное описание столбцов представлено на рисунке 6.

<i>APIEndPoint</i>		<i>endpoint of api</i>						
#	name	type	size	default	primary	foreign	unique	description
1	endpoint	text	-		no	-	no	api endpoint

Рисунок 6 – Описание столбцов таблицы APIEndPoint

Интерфейс приложения

Приложение состоит из 2 форм:

1. Main Form: служит для отображения карты и изменения масштаба, а также перехода к форме настроек.
2. Settings Form: форма настроек, для задания конечно точки API, времени кэширования, а также выбора активных дополнительных слоев и их цветов.

Форма Main Form

На рисунке 7 показан макет внешнего вида основной формы.

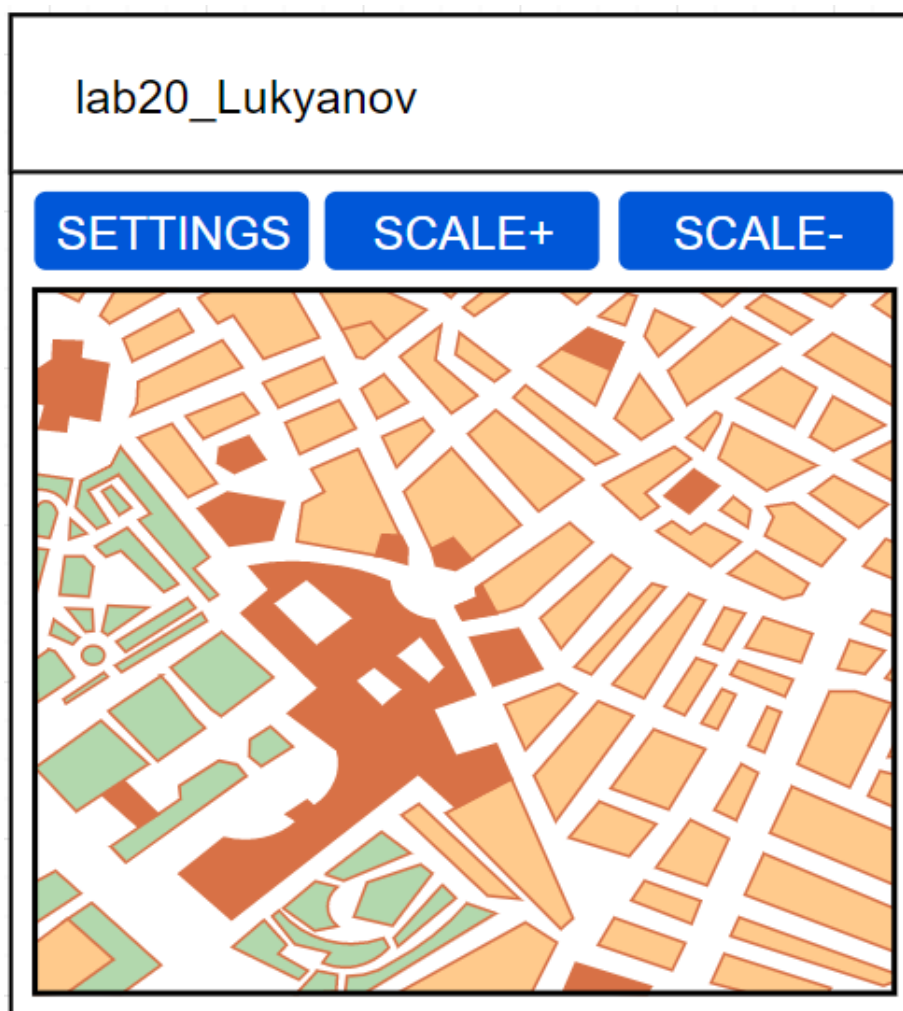


Рисунок 7 – Макет формы Main Form

На рисунке 8 показан внешний вид основной формы в приложении.



Рисунок 8 – Форма Main Form в приложении

Форма Settings Form

На рисунке 9 показан макет внешнего вида формы настроек.

lab20_Lukyanov

RETURN

API:

SAVE

Cache Time:

SAVE

CLEAR CACHE

☒
Coastline

colors ▼

☒
River

colors ▼

☒
Road

colors ▼

☒
Railroad

colors ▼

SAVE

Рисунок 9 – Макет формы Settings Form

На рисунке 10 показан внешний вид основной формы в приложении.

lab20_lukyanov			
RETURN			
API:	<input type="text" value="http://tilemap.spbcoit.ru:7000"/>		SAVE
Cache time:	<input type="text" value="5d"/>		SAVE
CLEAR CACHE			
<input checked="" type="checkbox"/>	Coastline	BLACK	▼
<input type="checkbox"/>	River	BLUE	▼
<input checked="" type="checkbox"/>	Road	YELLOW	▼
<input type="checkbox"/>	Railroad	GREEN	▼
SAVE			

Рисунок 10 – Форма Settings Form в приложении

Описание протокола взаимодействия

Для взаимодействия с базой данных мобильное приложение использует API на основе хранимых процедур и сервера, который организует вызов процедур по протоколу HTTP и обмен данными.

Всего доступно 6 функций, список которых представлен на рисунке 11.

1. `/raster` – получение уровней масштабирования с характеристиками.
2. `/raster/level/{x}-{y}` – получение изображения тайла.
3. `/coastline/{level}` – получение списка точек для рисования берегов.
4. `/river/{level}` – получение списка точек для рисования рек.
5. `/road/{level}` – получение списка точек для рисования дорог.
6. `/railroad/{level}` – получение списка точек для рисования железных дорог.

raster map tiles		^
GET	<code>/raster</code>	available scale levels with parameters
GET	<code>/raster/{level}/{x}-{y}</code>	fetch raster tile
physical physical data		^
GET	<code>/coastline/{level}</code>	fetch coastline shapes
GET	<code>/river/{level}</code>	fetch river shapes
cultural		^
GET	<code>/road/{level}</code>	fetch road shapes
GET	<code>/railroad/{level}</code>	fetch railroad shapes

Рисунок 11 – Список функций

1. Растровые

Данный набор функций позволяет получить данные о тайлах карты.

1.1. Функция `/raster`

Обеспечивает возможность получения сведений о уровнях масштабирования карты.

Входных параметров нет.

Выходными параметрами будет массив уровней со следующими характеристиками:

— `level` – уровень масштабирования.

- `xtiles` – количество тайлов по горизонтали.
- `ytiles` – количество тайлов по вертикали.
- `width` – ширина тайлов.
- `height` – высота тайлов.
- `resolution` – коэффициент разрешения.

Пример вызова функции показан на рисунке 12.

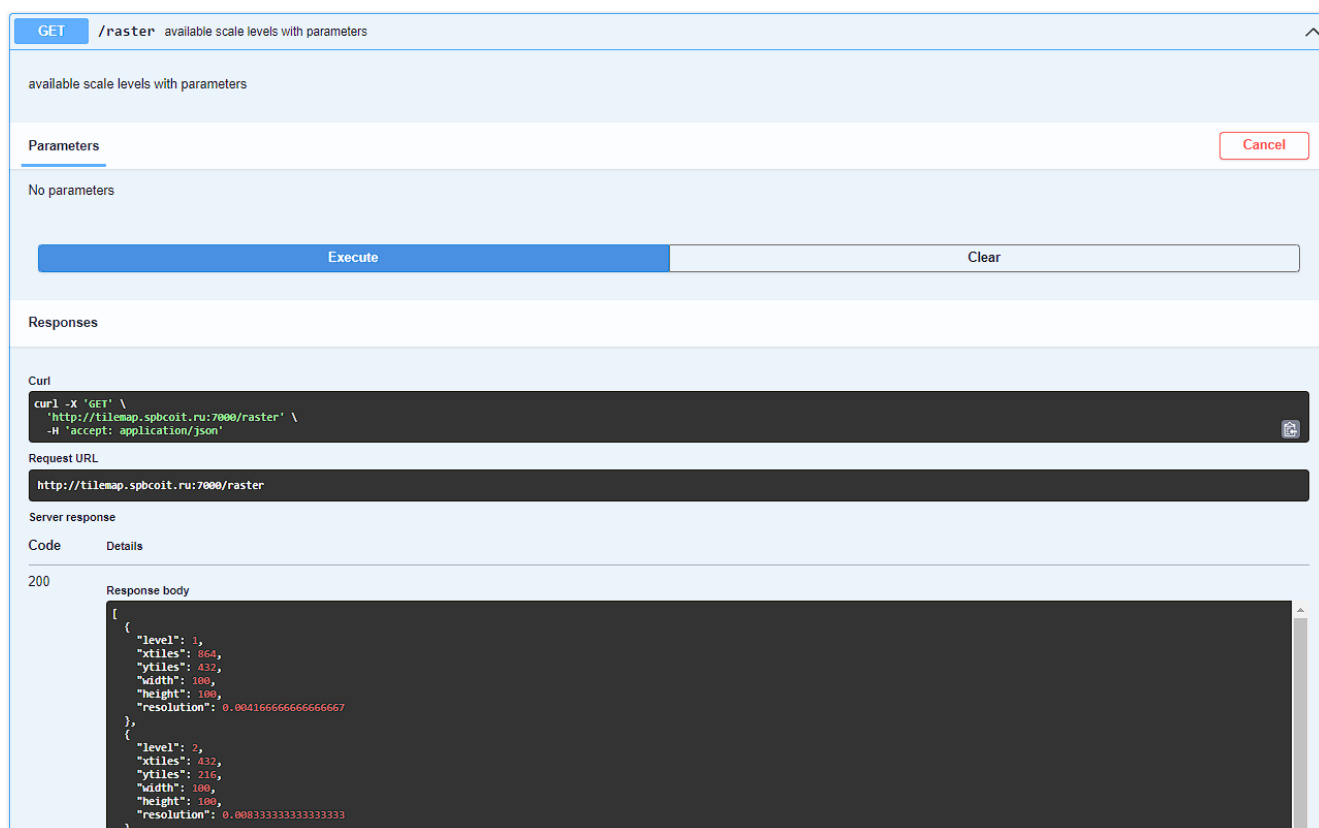


Рисунок 12 – Вызов функции /raster

1.2. Функция /raster/level/{x}-{y}

Обеспечивает возможность получения параметров тайла с использованием его координат и уровня масштабирования.

Входные параметры:

- `level` – уровень масштабирования.
- `x` – координата по горизонтали.
- `y` – координата по вертикали.

Выходные параметры:

- `lat` – ширина.

— data – изображение в формате текста.

[illegible]

2. Физические

2.1. Функция /coastline/{level}

Обеспечивает возможность получения точек для рисования линий берегов.

Входные параметры:

- level – уровень масштабирования.
- lat0 – левая-верхняя ширина.
- lon0 – левая-верхняя долгота.

— lat1 – правая-нижняя ширина.

— lon1 – правая-нижняя долгота.

Выходные параметры:

— массив массивов точек по горизонтали и вертикали.

Пример вызова функции показан на рисунке 14.

GET /coastline/{level} fetch coastline shapes

fetch coastline shapes

Parameters

Cancel

Name	Description
level * required (path)	scale level
lat0 * required (query)	top-left corner latitude
lon0 * required (query)	top-left corner longitude
lat1 * required (query)	bottom-right corner latitude
lon1 * required (query)	bottom-right corner longitude

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://tilemap.spbcoit.ru:7000/coastline/1?lat0=-10&lon0=10&lat1=10&lon1=-10' \
  -H 'accept: application/json'
```

Request URL

```
http://tilemap.spbcoit.ru:7000/coastline/1?lat0=-10&lon0=10&lat1=10&lon1=-10
```

Server response

Code Details

200

Response body

```
[
  {
    "x": 4.641956316000087,
    "y": 6.21882280000042
  },
  {
    "x": 4.717051629000082,
    "y": 6.1636416690000715
  },
  {
    "x": 4.74781334700009,
    "y": 6.138861395000049
  },
  {
    "x": 4.782725457000083,
```

Рисунок 14 – Вызов функции /coastline/{level}

2.2. Функция /river/{level}

Обеспечивает возможность получения точек для рисования линий рек.

Входные параметры:

— level – уровень масштабирования.

— lat0 – левая-верхняя ширина.

— lon0 – левая-верхняя долгота.

— lat1 – правая-нижняя ширина.

— lon1 – правая-нижняя долгота.

Выходные параметры:

— массив массивов точек по горизонтали и вертикали.

Пример вызова функции показан на рисунке 15.

GET /river/{level} fetch river shapes

fetch river shapes

Parameters Cancel

Name	Description
level * required (path)	scale level
lat0 * required (query)	top-left corner latitude
lon0 * required (query)	top-left corner longitude
lat1 * required (query)	bottom-right corner latitude
lon1 * required (query)	bottom-right corner longitude

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://tilemap.spbcoit.ru:7000/river/?lat0=-10&lon0=10&lat1=10&lon1=-10' \
-H 'accept: application/json'
```

Request URL

```
http://tilemap.spbcoit.ru:7000/river/?lat0=-10&lon0=10&lat1=10&lon1=-10
```

Server response

Code	Details
200	Response body

```
[
  {
    "x": -6.944433592749977,
    "y": 9.38886139530662
  },
  {
    "x": -6.810886571874977,
    "y": 9.55744049686912
  },
  {
    "x": -6.65288859374989,
    "y": 9.4933535280662
  },
  {
    "x": -6.549568546875,
    "y": 9.56983717655662
  }
]
```

Рисунок 15 – Вызов функции /river/{level}

3. Рукотворные

Данный набор функций позволяет получить рукотворные географические данные.

3.1. Функция /road/{level}

Обеспечивает возможность получения точек для рисования линий дорог.

Входные параметры:

- level – уровень масштабирования.
- lat0 – левая-верхняя ширина.
- lon0 – левая-верхняя долгота.
- lat1 – правая-нижняя ширина.
- lon1 – правая-нижняя долгота.

Выходные параметры:

- массив массивов точек по горизонтали и вертикали.

Пример вызова функции показан на рисунке 16.

GET /road/{level} fetch road shapes

fetch road shapes

Parameters

Cancel

Name	Description
level * required (path)	scale level
lat0 * required (query)	top-left corner latitude
lon0 * required (query)	top-left corner longitude
lat1 * required (query)	bottom-right corner latitude
lon1 * required (query)	bottom-right corner longitude

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://tilemap.sphocit.ru:7000/road/1?lat0=-10&lon0=10&lat1=10&lon1=-10' \
-H 'accept: application/json'
```

Request URL

```
http://tilemap.sphocit.ru:7000/road/1?lat0=-10&lon0=10&lat1=10&lon1=-10
```

Server response

Code	Details
200	Response body

```
[
  {
    "x": -5.164600581890028,
    "y": 9.31310001000124
  },
  {
    "x": 1.046099266597816,
    "y": 9.95470091191614
  },
  {
    "x": 1.0070410007607913,
    "y": 9.005010182439721
  }
]
```

Рисунок 16 – Вызов функции /road/{level}

3.2. Функция /railroad/{level}

Обеспечивает возможность получения точек для рисования линий железных дорог.

Входные параметры:

- level – уровень масштабирования.
- lat0 – левая-верхняя ширина.
- lon0 – левая-верхняя долгота.
- lat1 – правая-нижняя ширина.
- lon1 – правая-нижняя долгота.

Выходные параметры:

- массив массивов точек по горизонтали и вертикали

Пример вызова функции показан на рисунке 17.

The screenshot shows a REST client interface with the following sections:

- GET /railroad/{level} fetch railroad shapes**
- Parameters** (with a Cancel button):
 - level** (required, path): scale level, value: 1
 - lat0** (required, query): top-left corner latitude, value: -5
 - lon0** (required, query): top-left corner longitude, value: 5
 - lat1** (required, query): bottom-right corner latitude, value: 5
 - lon1** (required, query): bottom-right corner longitude, value: -5
- Execute** (blue button) and **Clear** (button)
- Responses** section:
 - Curl**:

```
curl -X 'GET' \
'http://tilemap.spbcnrt.ru:7000/railroad/1?lat0=-5&lon0=5&lat1=5&lon1=-5' \
-H 'accept: application/json'
```
 - Request URL**:

```
http://tilemap.spbcnrt.ru:7000/railroad/1?lat0=-5&lon0=5&lat1=5&lon1=-5
```
 - Server response**:
 - Code**: 200
 - Details**
 - Response body**:

```
[
  {
    "x": -1.7269438888888888,
    "y": 4.993655555555555
  },
  {
    "x": -1.7252774999999999,
    "y": 4.970555277777778
  },
  {
    "x": -1.7257222222222222,
    "y": 4.947734722222222
  },
  {
    "x": -1.7269438888888888,
    "y": 4.993655555555555
  }
]
```

Рисунок 17 – Вызов функции /railroad/{level}

Демонстрация работы приложения

Изображения карты отображаются (рис. 18):

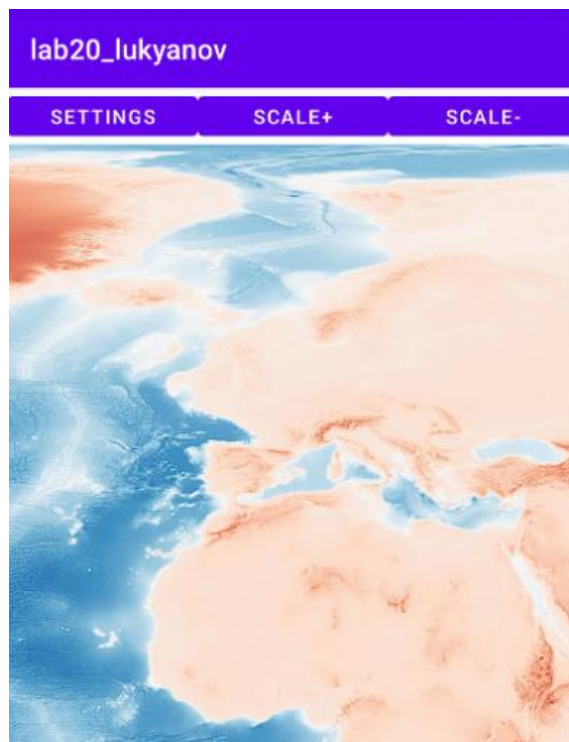


Рисунок 18 – Отображение карты

Реализовано перемещение по карте с загрузкой соответствующих тайлов (рис. 19):

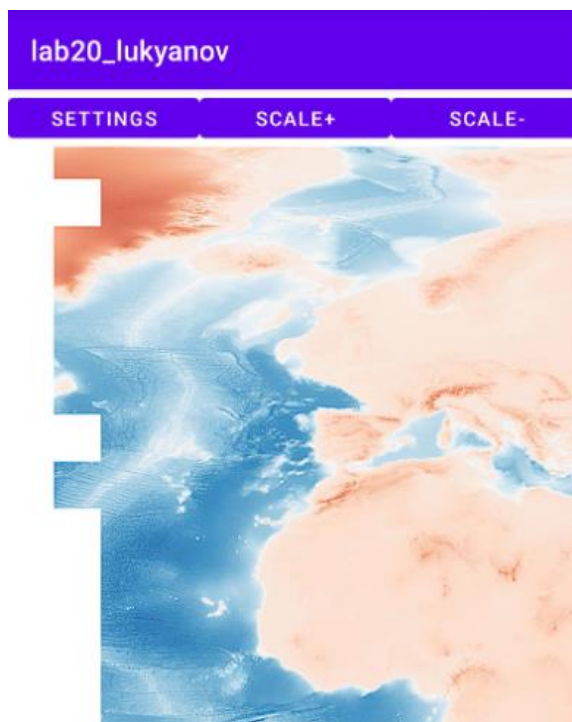


Рисунок 19 – Перемещение по карте

Имеется возможность увеличения и уменьшение масштаба, также при перезапуске приложения показывается последний уровень масштабирования и область обзора карты (рис. 20):

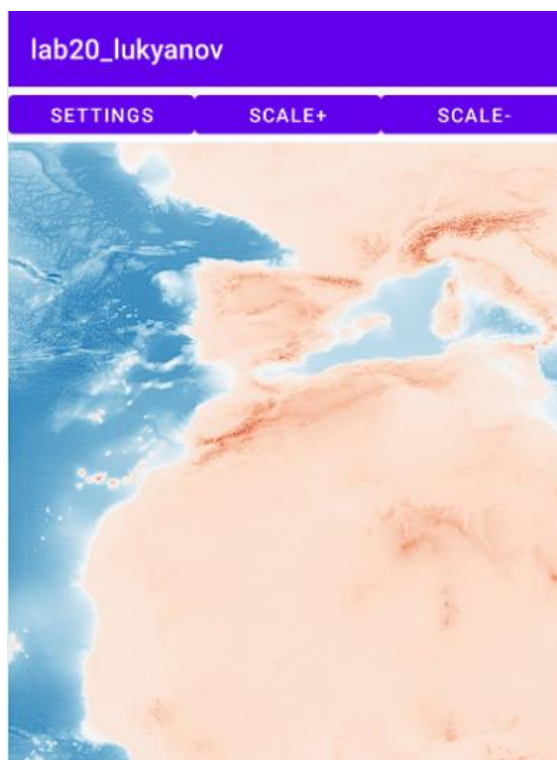


Рисунок 20 – Изменение масштаба

Имеется возможность настройки, а именно: задания точки API и времени кэширования, а также дополнительных слоев отображения (рис. 21):

lab20_lukyanov		
RETURN		
API:	<input type="text" value="http://tilemap.spbcoit.ru:7000"/>	SAVE
Cache time:	<input type="text" value="5d"/>	SAVE
CLEAR CACHE		
<input type="checkbox"/> Coastline	BLACK	▼
<input type="checkbox"/> River	BLUE	▼
<input type="checkbox"/> Road	YELLOW	▼
<input type="checkbox"/> Railroad	BLUE	▼
SAVE		

Рисунок 21 – Настройки

Изменим конечную точку API (рис. 22):



The screenshot shows a web interface for configuring an API endpoint. At the top, there is a text input field containing the username 'lab20_lukyanov'. Below it is a blue button labeled 'RETURN'. Underneath the button is the label 'API:' followed by a text input field containing the URL 'ttp://tilemap.spbcoit.ru:7000'. To the right of this field is a blue button labeled 'SAVE'.

Рисунок 22 – Изменение конечной точки API

Теперь проверим загрузку новых тайлов (рис. 23):

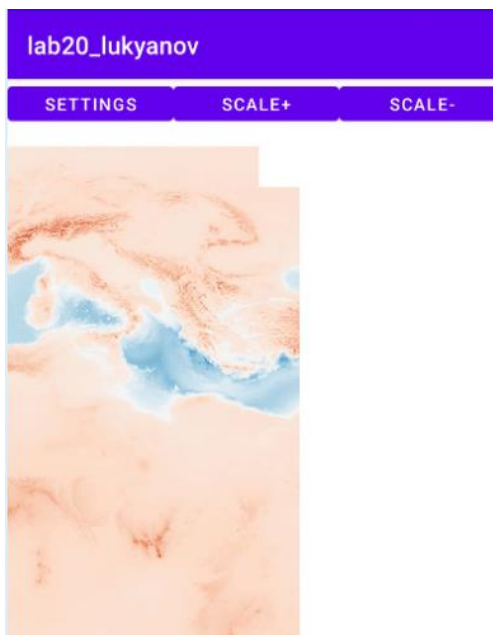
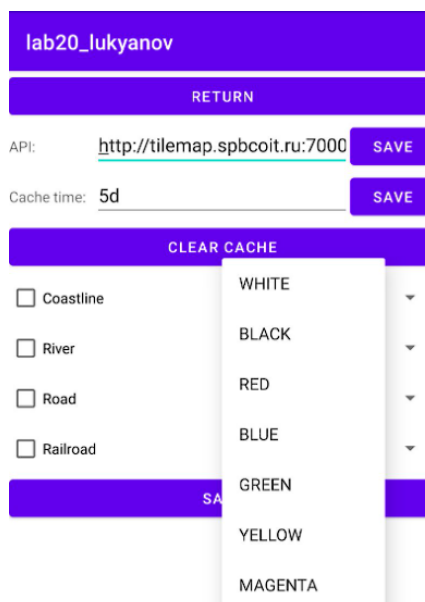


Рисунок 23 – Загрузка тайлов по новой конечной точке API

Имеется возможность настроить дополнительные слои отображения и задания им цветов (рис. 24):



The screenshot shows a web interface for configuring map layers. At the top, there is a text input field containing the username 'lab20_lukyanov'. Below it is a blue button labeled 'RETURN'. Underneath the button is the label 'API:' followed by a text input field containing the URL 'http://tilemap.spbcoit.ru:7000'. To the right of this field is a blue button labeled 'SAVE'. Below the API field is the label 'Cache time:' followed by a text input field containing '5d'. To the right of this field is a blue button labeled 'SAVE'. Below the cache time field is a blue button labeled 'CLEAR CACHE'. Below the 'CLEAR CACHE' button is a list of checkboxes for map layers: 'Coastline', 'River', 'Road', and 'Railroad'. To the right of the checkboxes is a dropdown menu with a list of colors: 'WHITE', 'BLACK', 'RED', 'BLUE', 'GREEN', 'YELLOW', and 'MAGENTA'. The 'SA' button is partially visible at the bottom.

Рисунок 24 – Список доступных цветов

Выберем все слои и зададим каждому свой цвет (рис. 25):

lab20_lukyanov

RETURN

API: SAVE

Cache time: SAVE

CLEAR CACHE

<input checked="" type="checkbox"/>	Coastline	BLACK	▼
<input checked="" type="checkbox"/>	River	BLUE	▼
<input checked="" type="checkbox"/>	Road	YELLOW	▼
<input checked="" type="checkbox"/>	Railroad	RED	▼

SAVE

Рисунок 25 – Задание дополнительных слоев и их цветов

Теперь посмотрим их отображение на карте (рис. 26):



Рисунок 26 – Отображение дополнительных слоев

Приложение имеет собственную иконку (рис. 27):



Рисунок 27 – Иконка приложения