Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/05 Интеллектуальный анализ больших данных в системах поддержки принятия решений.**

О Т Ч Е Т

по домашнему заданию № 1

Название: Дескриптивный анализ данных

Дисциплина: Методы машинного обучения

Студент	ИУ6-21М (Группа)	(Подпись, дата)	Д.С. Голубятников (И.О. Фамилия)
Преподаватель		(Подпись, дата)	С.Ю. Папулин (И.О. Фамилия)

Москва, 2024

In [1]:
```
surname = "Голубятников"  # Ваша фамилия

alp = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'
w = [1, 42, 21, 21, 34,  6, 44, 26, 18, 44, 38, 26, 14, 43,  4, 49, 45,
        7, 42, 29,  4,  9, 36, 34, 31, 29,  5, 30,  4, 19, 28, 25, 33]

d = dict(zip(alp, w))
variant =  sum([d[el] for el in surname.lower()]) % 40 + 1

print("Задача № 1, шаг 5 - вариант: ", variant % 5 + 1)
print("Задача № 1, шаг 11 - вариант: ", variant % 2 + 1 )
print("задача № 2 - вариант: ", variant % 4 + 1)
```

```
Задача № 1, шаг 5 - вариант:  3
Задача № 1, шаг 11 - вариант:  2
задача № 2 - вариант:  2
```

Задание 1. Анализ индикаторов качества государственного управления (The Worldwide Government Indicators, WGI)

Импорт библиотек

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.cm as cm
import matplotlib.pyplot as plt
from matplotlib import gridspec
```

1. Загрузка данных в DataFrame

In [3]:
```
page_number = 6
df = pd.read_excel('wgidataset.xlsx', sheet_name=page_number, header = 14)
df
```

Out[3]:

	Country/Territory	Code	Estimate	StdErr	NumSrc	Rank	Lower	Upper	Estin
0	Aruba	ABW	NaN	NaN	NaN	NaN	NaN	NaN	
1	Andorra	ADO	1.318143	0.480889	1.0	87.096771	72.043015	96.774193	1.3
2	Afghanistan	AFG	-1.291705	0.340507	2.0	4.301075	0.000000	27.419355	-1.1
3	Angola	AGO	-1.167702	0.262077	4.0	9.677420	0.537634	27.419355	-1.1
4	Anguilla	AIA	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	
209	Serbia	SRB	-1.140072	0.262077	4.0	11.827957	0.537634	29.032259	-1.1
210	South Africa	ZAF	0.732927	0.210325	6.0	76.344086	66.129036	81.182793	0.6
211	Congo, Dem. Rep.	ZAR	-1.647852	0.315914	3.0	0.000000	0.000000	12.365591	-1.4
212	Zambia	ZMB	-0.840641	0.262077	4.0	24.731182	5.913979	41.397850	-0.8
213	Zimbabwe	ZWE	-0.278847	0.244907	5.0	47.849461	30.645161	60.752689	-0.5

214 rows × 146 columns

2. Датасет отсортированный по убыванию индекса

In [4]:
```
df_sorted = df.sort_index(ascending=False)
df_sorted
```

Out[4]:

	Country/Territory	Code	Estimate	StdErr	NumSrc	Rank	Lower	Upper	Estin
213	Zimbabwe	ZWE	-0.278847	0.244907	5.0	47.849461	30.645161	60.752689	-0.5
212	Zambia	ZMB	-0.840641	0.262077	4.0	24.731182	5.913979	41.397850	-0.8
211	Congo, Dem. Rep.	ZAR	-1.647852	0.315914	3.0	0.000000	0.000000	12.365591	-1.4
210	South Africa	ZAF	0.732927	0.210325	6.0	76.344086	66.129036	81.182793	0.6
209	Serbia	SRB	-1.140072	0.262077	4.0	11.827957	0.537634	29.032259	-1.1
...	...	...	...	...	...	...	...	...	
4	Anguilla	AIA	NaN	NaN	NaN	NaN	NaN	NaN	
3	Angola	AGO	-1.167702	0.262077	4.0	9.677420	0.537634	27.419355	-1.1
2	Afghanistan	AFG	-1.291705	0.340507	2.0	4.301075	0.000000	27.419355	-1.1
1	Andorra	ADO	1.318143	0.480889	1.0	87.096771	72.043015	96.774193	1.3
0	Aruba	ABW	NaN	NaN	NaN	NaN	NaN	NaN	

214 rows × 146 columns

Выведем название страны, код и rank за 2022 год

In [5]:
```
df_2022 = df_sorted[['Country/Territory','Rank.23']]
df_2022
```

Out[5]:

	Country/Territory	Rank.23
213	Zimbabwe	8.490566
212	Zambia	34.433964
211	Congo, Dem. Rep.	3.301887
210	South Africa	44.811321
209	Serbia	35.377357
...	...	...
4	Anguilla	88.679245
3	Angola	30.660378
2	Afghanistan	12.264151
1	Andorra	88.679245
0	Aruba	77.830185

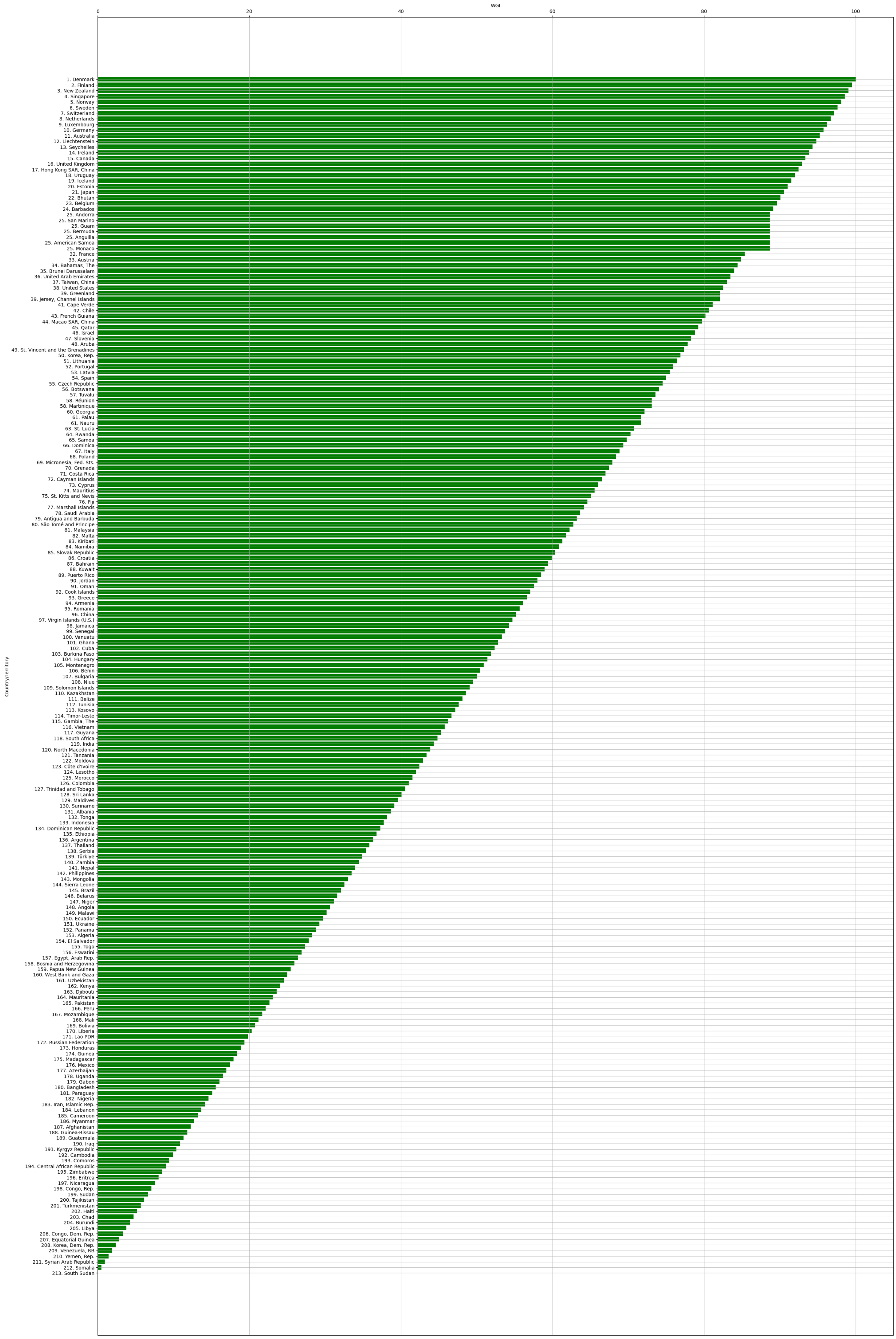
214 rows × 2 columns

3. Данные по индексу WGI за 2022 год в виде горизонтального столбчатого графика

In [6]:

```
df_sorted_by_wgi = df_2022.sort_values('Rank.23', ascending=True)

if df_sorted_by_wgi['Rank.23'].isnull().any():
    print("Найдены NaN значения в 'Rank.23'.")
    # удаление NaN
    df_sorted_by_wgi = df_sorted_by_wgi.dropna(subset=['Rank.23'])

# присвоение порядковых номеров
df_sorted_by_wgi['Num'] = df_sorted_by_wgi['Rank.23'].rank(method='min', ascending=

# формирование названия
df_sorted_by_wgi['Name'] = df_sorted_by_wgi['Num'].astype(str) + '. ' + df_sorted_b

# построение графика
plt.figure(figsize=(30, 50))  # размер графика
plt.barh(df_sorted_by_wgi['Name'], df_sorted_by_wgi['Rank.23'], color='green')


# вывод числовых значений справа от bar'a
# for index, value in enumerate(df_sorted_by_wgi['Rank.23']):
#     plt.text(value, index, str(value), va='center', ha='left', fontsize=10)

plt.gca().xaxis.set_ticks_position('top')
plt.gca().xaxis.set_label_position('top')

plt.grid(axis='x', linestyle='-', alpha=0.7)
plt.grid(axis='y', linestyle='-', alpha=0.7)

plt.xlabel('WGI')
plt.ylabel('Country/Territory')
plt.show()
```

```
Найдены NaN значения в 'Rank.23'.
```



4. Получение списка стран входящего в регион Europe and Central Asia из датасета regions

In [7]:

```python
df_by_region = pd.read_excel('regions.xlsx')
df_by_region_ECA = df_by_region.loc[df_by_region['Region']=='ECA']

df_by_region_ECA
```

Out[7]:

	Country	Code	Region
1	Albania	ALB	ECA
5	Armenia	ARM	ECA
8	Azerbaijan	AZE	ECA
13	Belarus	BLR	ECA
18	Bosnia and Herzegovina	BIH	ECA
59	Georgia	GEO	ECA
83	Kazakhstan	KAZ	ECA
87	Kosovo	KSV	ECA
89	Kyrgyzstan	KGZ	ECA
107	Moldova	MDA	ECA
109	Montenegro	MNE	ECA
120	North Macedonia	MKD	ECA
133	Russia	RUS	ECA
140	Serbia	SRB	ECA
158	Tajikistan	TJK	ECA
165	Turkey	TUR	ECA
166	Turkmenistan	TKM	ECA
168	Ukraine	UKR	ECA
173	Uzbekistan	UZB	ECA

5. Получекние стран из WGI входящих в указанный регион

In [8]:
```
df_sorted_by_ECA_by_name = df[df['Country/Territory'].isin(df_by_region_ECA['Countr
df_sorted_by_ECA_by_name
```

Out[8]:

	Country/Territory	Code	Estimate	StdErr	NumSrc	Rank	Lower	Upper	Estin
5	Albania	ALB	-0.893903	0.315914	3.0	19.354839	2.688172	43.010754	-0.9
9	Armenia	ARM	-0.473051	0.340507	2.0	38.172043	15.053763	59.139786	-0.9
14	Azerbaijan	AZE	-1.445619	0.275614	3.0	2.688172	0.000000	17.204302	-1.2
23	Bosnia and Herzegovina	BIH	-0.270570	0.275614	3.0	48.924732	28.494623	60.752689	-0.4
24	Belarus	BLR	-0.389609	0.340507	2.0	42.473118	17.741936	60.752689	-0.3
69	Georgia	GEO	-1.527264	0.340507	2.0	1.075269	0.000000	17.741936	-0.9
98	Kazakhstan	KAZ	-1.132820	0.275614	3.0	12.365591	0.537634	30.107527	-1.0
117	Kosovo	KSV	NaN	NaN	NaN	NaN	NaN	NaN	
121	Moldova	MDA	-0.437427	0.275614	3.0	39.784946	19.354839	57.526882	-0.3
126	North Macedonia	MKD	-0.613846	0.275614	3.0	32.258064	14.516129	51.075268	-0.6
131	Montenegro	MNE	NaN	NaN	NaN	NaN	NaN	NaN	0.4
186	Tajikistan	TJK	-1.273033	0.340507	2.0	5.913979	0.000000	29.032259	-1.2
187	Turkmenistan	TKM	-1.021493	0.340507	2.0	15.591398	0.537634	38.709679	-1.0
197	Ukraine	UKR	-1.110137	0.255844	4.0	13.440860	0.537634	29.569893	-1.2
200	Uzbekistan	UZB	-1.128821	0.320799	2.0	12.903226	0.000000	32.258064	-1.1
209	Serbia	SRB	-1.140072	0.262077	4.0	11.827957	0.537634	29.032259	-1.1

16 rows × 146 columns

In [9]:

```
# или так
df_sorted_by_ECA = df[df['Code'].isin(df_by_region_ECA['Code'])]
df_sorted_by_ECA
```

Out[9]:

	Country/Territory	Code	Estimate	StdErr	NumSrc	Rank	Lower	Upper	Estin
5	Albania	ALB	-0.893903	0.315914	3.0	19.354839	2.688172	43.010754	-0.9
9	Armenia	ARM	-0.473051	0.340507	2.0	38.172043	15.053763	59.139786	-0.9
14	Azerbaijan	AZE	-1.445619	0.275614	3.0	2.688172	0.000000	17.204302	-1.2
23	Bosnia and Herzegovina	BIH	-0.270570	0.275614	3.0	48.924732	28.494623	60.752689	-0.4
24	Belarus	BLR	-0.389609	0.340507	2.0	42.473118	17.741936	60.752689	-0.3
69	Georgia	GEO	-1.527264	0.340507	2.0	1.075269	0.000000	17.741936	-0.9
98	Kazakhstan	KAZ	-1.132820	0.275614	3.0	12.365591	0.537634	30.107527	-1.0
100	Kyrgyz Republic	KGZ	-0.993923	0.340507	2.0	17.204302	0.537634	39.784946	-1.0
117	Kosovo	KSV	NaN	NaN	NaN	NaN	NaN	NaN	
121	Moldova	MDA	-0.437427	0.275614	3.0	39.784946	19.354839	57.526882	-0.3
126	North Macedonia	MKD	-0.613846	0.275614	3.0	32.258064	14.516129	51.075268	-0.6
131	Montenegro	MNE	NaN	NaN	NaN	NaN	NaN	NaN	0.4
163	Russian Federation	RUS	-1.053342	0.210325	6.0	15.053763	2.688172	29.032259	-0.9
186	Tajikistan	TJK	-1.273033	0.340507	2.0	5.913979	0.000000	29.032259	-1.2
187	Turkmenistan	TKM	-1.021493	0.340507	2.0	15.591398	0.537634	38.709679	-1.0
192	Türkiye	TUR	-0.148074	0.210325	6.0	51.612904	36.559139	61.827957	-0.3
197	Ukraine	UKR	-1.110137	0.255844	4.0	13.440860	0.537634	29.569893	-1.2
200	Uzbekistan	UZB	-1.128821	0.320799	2.0	12.903226	0.000000	32.258064	-1.1
209	Serbia	SRB	-1.140072	0.262077	4.0	11.827957	0.537634	29.032259	-1.1

19 rows × 146 columns

Получение WGI содержащего estimate по годам для заданного региона

In [13]:

```
columns = {
'Estimate' : '1996',
'Estimate.1' : '1998',
'Estimate.2' : '2000',
'Estimate.3' : '2002',
'Estimate.4' : '2003',
'Estimate.5' : '2004',
'Estimate.6' : '2005',
'Estimate.7' : '2006',
'Estimate.8' : '2007',
'Estimate.9' : '2008',
'Estimate.10' : '2009',
'Estimate.11' : '2010',
'Estimate.12' : '2011',
'Estimate.13' : '2012',
'Estimate.14' : '2013',
'Estimate.15' : '2014',
'Estimate.16' : '2015',
'Estimate.17' : '2016',
'Estimate.18' : '2017',
'Estimate.19' : '2018',
```

```
'Estimate.20' : '2019',
'Estimate.21' : '2020',
'Estimate.22' : '2021',
'Estimate.23' : '2022',
}

df_sorted_by_ECA_wgi = df_sorted_by_ECA.filter(like='Estimate')
df_sorted_by_ECA_countries = df_sorted_by_ECA['Country/Territory']
df_sorted_by_ECA_wgi_estimate = df_sorted_by_ECA_wgi
df_sorted_by_ECA_wgi_estimate = pd.concat([df_sorted_by_ECA_countries,df_sorted_by_
df_sorted_by_ECA_wgi_estimate_output = df_sorted_by_ECA_wgi_estimate.rename(columns
'Estimate' : '1996',
'Estimate.1' : '1998',
'Estimate.2' : '2000',
'Estimate.3' : '2002',
'Estimate.4' : '2003',
'Estimate.5' : '2004',
'Estimate.6' : '2005',
'Estimate.7' : '2006',
'Estimate.8' : '2007',
'Estimate.9' : '2008',
'Estimate.10' : '2009',
'Estimate.11' : '2010',
'Estimate.12' : '2011',
'Estimate.13' : '2012',
'Estimate.14' : '2013',
'Estimate.15' : '2014',
'Estimate.16' : '2015',
'Estimate.17' : '2016',
'Estimate.18' : '2017',
'Estimate.19' : '2018',
'Estimate.20' : '2019',
'Estimate.21' : '2020',
'Estimate.22' : '2021',
'Estimate.23' : '2022',
})
df_sorted_by_ECA_wgi_estimate_output
```

Out[13]:

	Country/Territory	1996	1998	2000	2002	2003	2004	2005
5	Albania	-0.893903	-0.992025	-0.855564	-0.845341	-0.853787	-0.723732	-0.813264
9	Armenia	-0.473051	-0.936306	-0.848086	-0.778165	-0.649022	-0.703455	-0.683350
14	Azerbaijan	-1.445619	-1.289568	-1.292383	-1.195502	-1.046232	-1.180832	-1.052521
23	Bosnia and Herzegovina	-0.270570	-0.402136	-0.595849	-0.387543	-0.282097	-0.324706	-0.232809
24	Belarus	-0.389609	-0.323170	-0.504358	-0.657885	-0.537340	-0.759293	-0.723231
69	Georgia	-1.527264	-0.982506	-1.040173	-1.264142	-0.650767	-0.459568	-0.212854
98	Kazakhstan	-1.132820	-1.078855	-1.149889	-1.113585	-1.026524	-1.105954	-1.014963
100	Kyrgyz Republic	-0.993923	-1.058797	-1.007479	-1.064770	-1.032819	-1.097712	-1.267269
117	Kosovo	NaN	NaN	0.371275	0.360482	-0.504248	-0.305081	-0.526772
121	Moldova	-0.437427	-0.399806	-0.623196	-0.980350	-0.884173	-1.039576	-0.672626
126	North Macedonia	-0.613846	-0.636420	-0.638976	-0.826058	-0.651164	-0.554222	-0.488401
131	Montenegro	NaN	0.497083	-0.177506	-0.156704	-0.411994	-0.467465	-0.354254
163	Russian Federation	-1.053342	-0.954374	-0.943414	-0.954848	-0.783092	-0.825626	-0.847121
186	Tajikistan	-1.273033	-1.252361	-1.265868	-1.174971	-1.147746	-1.319210	-1.176047
187	Turkmenistan	-1.021493	-1.068734	-1.077138	-1.054273	-1.015132	-1.270521	-1.347762
192	Türkiye	-0.148074	-0.354078	-0.258080	-0.570129	-0.209118	-0.191904	-0.035273
197	Ukraine	-1.110137	-1.258210	-1.110609	-1.091394	-0.966655	-0.979342	-0.740643
200	Uzbekistan	-1.128821	-1.134654	-1.080415	-0.964739	-0.914872	-1.068812	-1.229638
209	Serbia	-1.140072	-1.195605	-1.156671	-0.895785	-0.494189	-0.493294	-0.406051

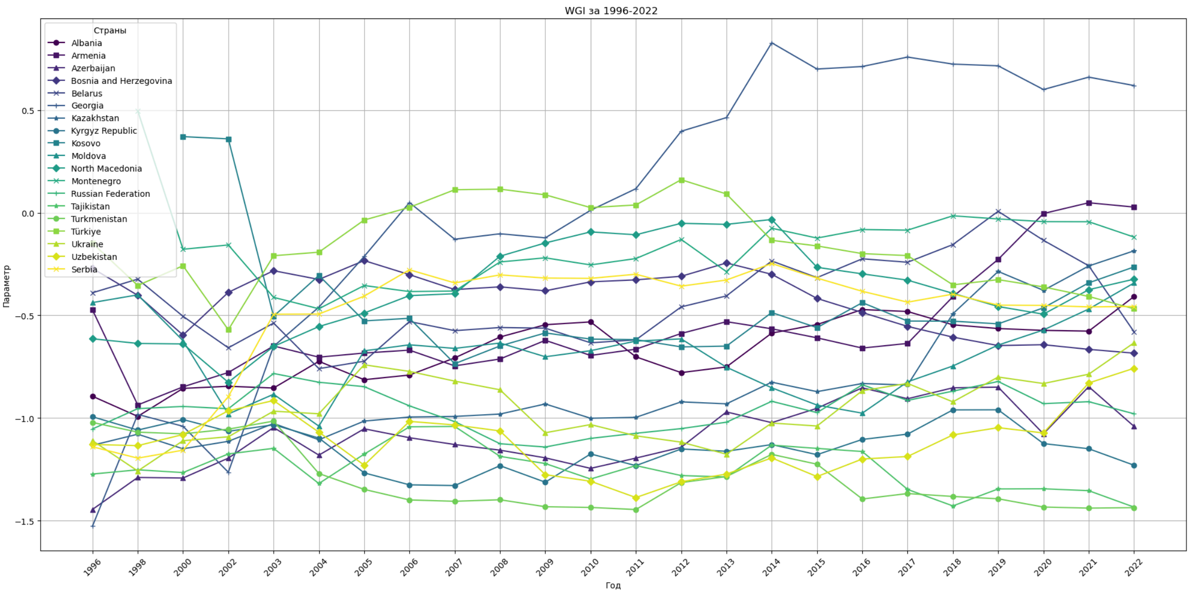
19 rows × 25 columns

6. Построение графика индекса WGI за 1996-2022 для Europe and Central Asia

In [11]:

```
# df_sorted_by_ECA_wgi = df_sorted_by_ECA_wgi
# df_sorted_by_ECA_wgi
# plt.figure(figsize=(30, 50))
# plt.title(" WGI за 1996-2022 для региона ECA")
# plt.plot(df_sorted_by_ECA_wgi_estimate['Country/Territory'],
#          df_sorted_by_ECA_wgi)


# установим индекс для DataFrame
df_sorted_by_ECA_wgi_estimate_output_T = df_sorted_by_ECA_wgi_estimate_output.set_i

df_sorted_by_ECA_wgi_estimate_output_T = df_sorted_by_ECA_wgi_estimate_output.T


plt.figure(figsize=(20, 10))

markers = ['o', 's', '^', 'D', 'x', '+', '*']

# генерация уникальных цветов
colors = plt.cm.viridis(np.linspace(0, 1, len(df_sorted_by_ECA_wgi_estimate_output_
```

```python
for i, country in enumerate(df_sorted_by_ECA_wgi_estimate_output_T.columns):
    marker = markers[i % len(markers)]  # используем маркер по кругу
    plt.plot(df_sorted_by_ECA_wgi_estimate_output_T.index,
             df_sorted_by_ECA_wgi_estimate_output_T[country],
             marker=marker, color=colors[i], label=country)

plt.title("WGI за 1996-2022 ")
plt.xlabel('Год')
plt.ylabel('Параметр')
plt.xticks(rotation=45)  # поворачиваем метки по оси X для удобства чтения
plt.legend(title='Страны')
plt.grid()
plt.tight_layout()

plt.show()
```



7. Получение стран с наибольшим и наименьшим значением WGI в регионе Europe and Central Asia

In [15]:
```python
# надо перезапустить In [10] иначе не работает
df_sorted_by_ECA_wgi_estimate_output_2022 = df_sorted_by_ECA_wgi_estimate_output[['
df_sorted_by_ECA_wgi_estimate_output_2022
max_2022 = df_sorted_by_ECA_wgi_estimate_output_2022['2022'].max()
maxes = df_sorted_by_ECA_wgi_estimate_output_2022[df_sorted_by_ECA_wgi_estimate_out
print("Страны с наибольшим индексом WGI за 2022 год:")
maxes
```

```
Страны с наибольшим индексом WGI за 2022 год:
```

Out[15]:

	Country/Territory	2022
69	Georgia	0.620238

In [16]:
```python
min_2022 = df_sorted_by_ECA_wgi_estimate_output_2022['2022'].min()
mins = df_sorted_by_ECA_wgi_estimate_output_2022[df_sorted_by_ECA_wgi_estimate_outp
print("\nСтраны с наименьшим индексом WGI за 2022 год:")
mins
```

```
Страны с наименьшим индексом WGI за 2022 год:
```

Out[16]:

	Country/Territory	2022
187	Turkmenistan	-1.436774

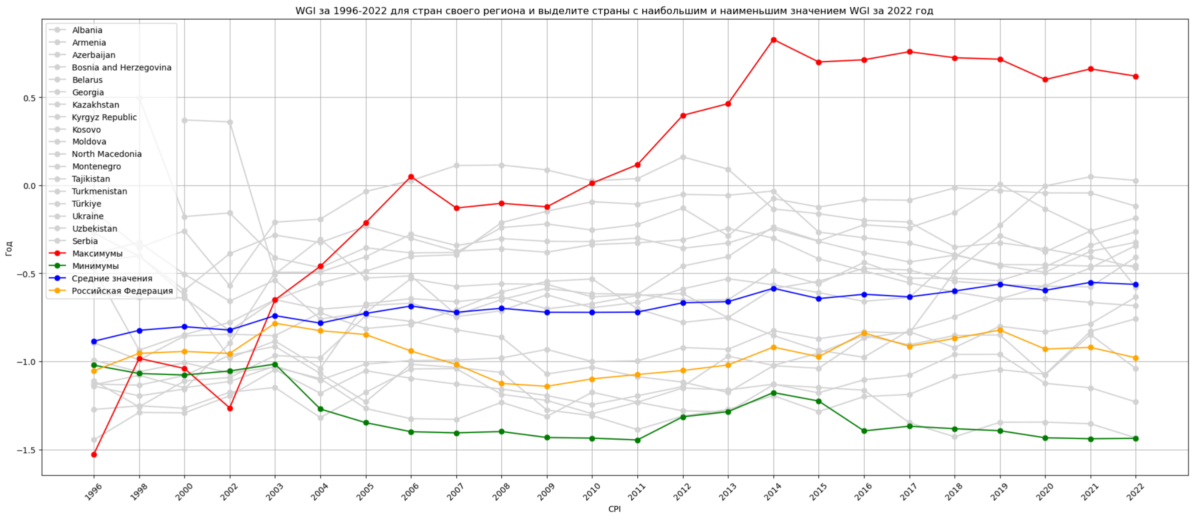
8. Определение среднего значения за каждый год в период с 1996 по 2022 в регионе Europe and Central Asia

In [17]:
```
means = df_sorted_by_ECA_wgi_estimate_output.drop(columns = ['Country/Territory'])
means = df_sorted_by_ECA_wgi_estimate_output.mean(numeric_only=True)
means
```

Out[17]:
```
1996   -0.885471
1998   -0.823362
2000   -0.802862
2002   -0.821879
2003   -0.740051
2004   -0.782648
2005   -0.727624
2006   -0.685522
2007   -0.721019
2008   -0.698221
2009   -0.720827
2010   -0.721366
2011   -0.719957
2012   -0.666920
2013   -0.660700
2014   -0.584655
2015   -0.643372
2016   -0.618599
2017   -0.633326
2018   -0.600420
2019   -0.561316
2020   -0.596217
2021   -0.551040
2022   -0.562725
dtype: float64
```

9. Построение графика индекса WGI за 1996-2022 для стран региона Europe and Central Asia с выделением страны с наибольшим и наименьшим значением WGI за 2022 год, а также отображение среднего значения по региону и РФ

In [18]:
```
fig, ax = plt.subplots(figsize=(25, 10))
task1_9 = df_sorted_by_ECA_wgi_estimate_output_T
task1_9 = task1_9.drop(columns=['Russian Federation'])

# строим основной график
task1_9.plot(ax=ax, color='lightgrey', marker='o', legend=False, title="WGI за 1996

# добавляем максимумы, минимумы и средние значения
ax.plot(df_sorted_by_ECA_wgi_estimate_output_T[maxes['Country/Territory']], color='
ax.plot(df_sorted_by_ECA_wgi_estimate_output_T[mins['Country/Territory']], color='g
ax.plot(means, color='blue', marker='o', label='Средние значения')

# добавляем линию для "Russian Federation"
ax.plot(df_sorted_by_ECA_wgi_estimate_output_T['Russian Federation'], color='orange

ax.set_xticks(df_sorted_by_ECA_wgi_estimate_output_T.index)  # Установка всех годов
ax.set_xticklabels(df_sorted_by_ECA_wgi_estimate_output_T.index, rotation=45)  # По


ax.set_xlabel("CPI")
ax.set_ylabel("Год")

ax.legend()
ax.grid()  # Сетка по каждому году на оси X
plt.show()
```



11. Определение изменения rank с 1996 по 2022 в регионе Americas

In [19]:
```
df_by_region_AME = df_by_region.loc[df_by_region['Region']=='AME']

df_by_region_AME
```

Out[19]:

	Country	Code	Region
4	Argentina	ARG	AME
9	Bahamas	BHS	AME
12	Barbados	BRB	AME
17	Bolivia	BOL	AME
20	Brazil	BRA	AME
27	Canada	CAN	AME
30	Chile	CHL	AME
32	Colombia	COL	AME
35	Costa Rica	CRI	AME
38	Cuba	CUB	AME
44	Dominica	DMA	AME
45	Dominican Republic	DOM	AME
46	Ecuador	ECU	AME
48	El Salvador	SLV	AME
63	Grenada	GRD	AME
64	Guatemala	GTM	AME
67	Guyana	GUY	AME
68	Haiti	HTI	AME
69	Honduras	HND	AME
80	Jamaica	JAM	AME
106	Mexico	MEX	AME
117	Nicaragua	NIC	AME
124	Panama	PAN	AME
126	Paraguay	PRY	AME
127	Peru	PER	AME
135	Saint Lucia	LCA	AME
136	Saint Vincent and the Grenadines	VCT	AME
153	Suriname	SUR	AME
163	Trinidad and Tobago	TTO	AME
171	United States of America	USA	AME
172	Uruguay	URY	AME
175	Venezuela	VEN	AME

In [20]:

```
df_sorted_by_AME = df[df['Code'].isin(df_by_region_AME['Code'])]
df_sorted_by_AME
```

Out[20]:

	Country/Territory	Code	Estimate	StdErr	NumSrc	Rank	Lower	Upper	Esti
8	Argentina	ARG	-0.101317	0.210325	6.0	53.763439	38.709679	62.903225	-0.
22	Bahamas, The	BHS	1.156810	0.418145	2.0	83.870964	69.892471	93.010750	1.
27	Bolivia	BOL	-0.824829	0.262077	4.0	25.268818	6.451613	41.935482	-0.
28	Brazil	BRA	-0.018580	0.210325	6.0	56.989246	44.086021	63.440861	0.
29	Barbados	BRB	1.542726	0.340507	2.0	90.860214	81.182793	96.774193	1.
35	Canada	CAN	2.031408	0.210325	6.0	96.236557	91.397850	100.000000	2.
37	Chile	CHL	1.454119	0.210325	6.0	90.322578	81.182793	93.010750	1.
43	Colombia	COL	-0.512254	0.210325	6.0	36.559139	23.118280	50.537636	-0.
46	Costa Rica	CRI	0.696142	0.244907	5.0	75.268814	63.440861	81.182793	0.
47	Cuba	CUB	0.289153	0.262077	4.0	63.440861	52.150539	75.806450	0.
53	Dominica	DMA	0.869897	0.480889	1.0	80.107529	59.139786	91.397850	0.
55	Dominican Republic	DOM	-0.422995	0.262077	4.0	41.397850	23.655914	57.526882	-0.
57	Ecuador	ECU	-0.684874	0.262077	4.0	30.107527	12.903226	49.462364	-0.
76	Grenada	GRD	0.869897	0.480889	1.0	80.107529	59.139786	91.397850	0.
78	Guatemala	GTM	-0.856944	0.244907	5.0	23.655914	6.451613	38.709679	-0.
81	Guyana	GUY	-0.140642	0.315914	3.0	52.688171	31.182796	65.053764	-0.
83	Honduras	HND	-1.078811	0.244907	5.0	14.516129	2.150538	30.645161	-1.
85	Haiti	HTI	-1.173277	0.315914	3.0	9.139785	0.000000	31.182796	-1.
95	Jamaica	JAM	0.187061	0.262077	4.0	61.827957	49.462364	73.118279	0.
110	St. Lucia	LCA	NaN	NaN	NaN	NaN	NaN	NaN	
124	Mexico	MEX	-0.512429	0.210325	6.0	36.021507	23.118280	50.537636	-0.
141	Nicaragua	NIC	-0.564544	0.244907	5.0	33.870968	17.741936	51.075268	-0.
150	Panama	PAN	-0.201106	0.262077	4.0	50.537636	31.720430	62.903225	-0.
152	Peru	PER	-0.398886	0.244907	5.0	41.935482	25.268818	57.526882	-0.
159	Paraguay	PRY	-1.166015	0.262077	4.0	10.215054	0.537634	27.419355	-1.
172	El Salvador	SLV	-0.865278	0.244907	5.0	21.505377	5.913979	38.709679	-0.
176	Suriname	SUR	0.185647	0.315914	3.0	61.290321	46.236561	75.268814	0.
190	Trinidad and Tobago	TTO	0.901226	0.262077	4.0	80.645164	69.892471	87.096771	0.
198	Uruguay	URY	1.124996	0.262077	4.0	82.258064	74.731186	90.860214	1.
199	United States	USA	1.571041	0.210325	6.0	91.397850	84.408600	94.086021	1.
201	St. Vincent and the Grenadines	VCT	NaN	NaN	NaN	NaN	NaN	NaN	
202	Venezuela, RB	VEN	-0.862947	0.210325	6.0	22.580645	7.526882	35.483871	-0.

32 rows × 146 columns

In [21]:
```
df_sorted_by_AME_rank = df_sorted_by_AME.filter(like='Rank')

df_sorted_by_AME_countries = df_sorted_by_AME['Country/Territory']
# df_sorted_by_AME_countries


df_sorted_by_AME_rank = pd.concat([df_sorted_by_AME_countries,df_sorted_by_AME_rank
df_sorted_by_AME_rank
df_sorted_by_AME_rank_output = df_sorted_by_AME_rank.rename(columns = {
'Rank' : '1996',
'Rank.1' : '1998',
'Rank.2' : '2000',
'Rank.3' : '2002',
'Rank.4' : '2003',
'Rank.5' : '2004',
'Rank.6' : '2005',
'Rank.7' : '2006',
'Rank.8' : '2007',
'Rank.9' : '2008',
'Rank.10' : '2009',
'Rank.11' : '2010',
'Rank.12' : '2011',
'Rank.13' : '2012',
'Rank.14' : '2013',
'Rank.15' : '2014',
'Rank.16' : '2015',
'Rank.17' : '2016',
'Rank.18' : '2017',
'Rank.19' : '2018',
'Rank.20' : '2019',
'Rank.21' : '2020',
'Rank.22' : '2021',
'Rank.23' : '2022',
})
df_sorted_by_AME_rank_output
```

Out[21]:

	Country/Territory	1996	1998	2000	2002	2003	2004	200
8	Argentina	53.763439	50.267380	52.127659	41.798943	40.211639	39.901478	43.41463
22	Bahamas, The	83.870964	85.561501	86.170212	89.417992	89.947090	91.625618	89.26829
27	Bolivia	25.268818	40.641712	40.425533	22.222221	23.280424	24.630543	24.39024
28	Brazil	56.989246	56.149734	57.978722	58.730160	59.788361	54.679802	51.70731
29	Barbados	90.860214	90.909088	90.957443	91.005295	91.005295	91.133003	90.73170
35	Canada	96.236557	95.187164	95.212769	94.179893	94.179893	92.118225	93.65853
37	Chile	90.322578	86.096260	91.489365	90.476189	83.597885	87.684731	91.21951
43	Colombia	36.559139	36.363636	42.553192	46.560848	50.793652	52.216747	52.19512
46	Costa Rica	75.268814	77.005348	74.468086	70.370369	72.486771	63.054188	65.36585
47	Cuba	63.440861	64.171120	65.957443	69.312172	62.433861	60.591133	62.92683
53	Dominica	80.107529	80.748665	81.382980	80.423279	79.365082	74.876846	74.63414
55	Dominican Republic	41.397850	29.411764	32.446808	39.682541	28.571428	32.019703	28.29268
57	Ecuador	30.107527	28.342245	27.659575	23.809525	29.100529	27.093596	25.36585
76	Grenada	80.107529	80.748665	81.382980	80.423279	79.365082	74.876846	74.63414
78	Guatemala	23.655914	26.203209	25.531916	33.333332	27.513227	30.049261	29.26829
81	Guyana	52.688171	46.524063	42.021278	47.089947	44.973545	40.886700	35.12195
83	Honduras	14.516129	16.577539	17.553192	15.873015	15.873015	18.719212	22.43902
85	Haiti	9.139785	9.090909	7.446808	0.000000	0.000000	2.463054	3.41463
95	Jamaica	61.827957	59.893047	59.042553	38.095238	41.269840	41.379311	46.34146
110	St. Lucia	NaN	NaN	NaN	NaN	NaN	62.561577	84.39024
124	Mexico	36.021507	43.850266	54.787235	47.619049	48.677250	41.871922	45.85365
141	Nicaragua	33.870968	20.855616	20.744680	39.153439	37.037037	38.423645	32.68292
150	Panama	50.537636	51.871658	45.744682	44.973545	43.386242	45.812809	42.92683
152	Peru	41.935482	41.176472	43.085106	46.031746	53.439152	38.916256	44.39024
159	Paraguay	10.215054	2.673797	1.063830	2.645503	3.174603	3.940887	4.87804
172	El Salvador	21.505377	25.668449	29.787233	25.925926	41.798943	35.960590	38.53658
176	Suriname	61.290321	62.566845	68.617020	65.079369	66.666664	61.576355	60.00000
190	Trinidad and Tobago	80.645164	73.796791	64.361702	58.201057	59.259258	55.665024	57.07317
198	Uruguay	82.258064	82.887703	82.446808	82.010582	82.539680	81.280785	81.95121
199	United States	91.397850	91.978607	92.021278	92.592590	92.063492	92.610840	91.70731
201	St. Vincent and the Grenadines	NaN	NaN	NaN	NaN	NaN	62.561577	82.43902
202	Venezuela, RB	22.580645	22.459892	32.978722	10.582010	10.052910	17.241379	16.09756

32 rows × 25 columns

In [22]:
```
df_sorted_by_AME_wgi_estimate_output_2022 = df_sorted_by_AME_rank_output[['Country/
df_sorted_by_AME_wgi_estimate_output_2022
AME_max_2022 = df_sorted_by_AME_rank_output['2022'].max()
AME_maxes = df_sorted_by_AME_wgi_estimate_output_2022[df_sorted_by_AME_wgi_estimate
print("Страны с наибольшим индексом WGI за 2022 год:")
country_with_max_value = AME_maxes['Country/Territory'].values[0]
AME_maxes
```

```
Страны с наибольшим индексом WGI за 2022 год:
```

Out[22]:

	Country/Territory	2022
35	Canada	93.396225

In [23]:
```
AME_min_2022 = df_sorted_by_AME_rank_output['2022'].min()
AME_mins = df_sorted_by_AME_wgi_estimate_output_2022[df_sorted_by_AME_wgi_estimate_
print("Страны с наименьшим индексом WGI за 2022 год:")
country_with_min_value = AME_mins['Country/Territory'].values[0]
AME_mins
```

```
Страны с наименьшим индексом WGI за 2022 год:
```

Out[23]:

	Country/Territory	2022
202	Venezuela, RB	1.886792

In [24]:
```
AME_means = df_sorted_by_AME_rank_output.drop(columns =['Country/Territory'])
AME_means = df_sorted_by_AME_rank_output.mean(numeric_only=True)
AME_means
```

Out[24]:
```
1996    53.279570
1998    52.655972
2000    53.581560
2002    51.587302
2003    51.728395
2004    51.200739
2005    52.728659
2006    53.033536
2007    53.200849
2008    53.929005
2009    53.588517
2010    53.497023
2011    54.028437
2012    51.836493
2013    51.747630
2014    48.332332
2015    48.660715
2016    48.854167
2017    48.229167
2018    47.961309
2019    48.050595
2020    46.964286
2021    45.982143
2022    45.592570
dtype: float64
```

In [25]:
```
# df_sorted_by_AME = df[df['Code'].isin(df_by_region_AME['Code'])]
# df_by_region
data = {'Country': df_sorted_by_AME_rank_output['Country/Territory'],
        '1996': df_sorted_by_AME_rank_output['1996'],
        '2022': df_sorted_by_AME_rank_output['2022'],
        'Rank_Difference': df_sorted_by_AME_rank_output['2022'] - df_sorted_by_AME_
rank_diff = pd.DataFrame(data)
rank_diff
```

Out[25]:

	Country	1996	2022	Rank_Difference
8	Argentina	53.763439	36.320755	-17.442684
22	Bahamas, The	83.870964	84.433960	0.562996
27	Bolivia	25.268818	20.754717	-4.514101
28	Brazil	56.989246	32.075470	-24.913776
29	Barbados	90.860214	89.150940	-1.709274
35	Canada	96.236557	93.396225	-2.840332
37	Chile	90.322578	80.660378	-9.662201
43	Colombia	36.559139	41.037735	4.478596
46	Costa Rica	75.268814	66.981133	-8.287682
47	Cuba	63.440861	52.358490	-11.082371
53	Dominica	80.107529	69.339622	-10.767906
55	Dominican Republic	41.397850	37.264153	-4.133698
57	Ecuador	30.107527	29.716982	-0.390545
76	Grenada	80.107529	67.452827	-12.654701
78	Guatemala	23.655914	11.320755	-12.335159
81	Guyana	52.688171	45.283020	-7.405151
83	Honduras	14.516129	18.867924	4.351794
85	Haiti	9.139785	5.188679	-3.951106
95	Jamaica	61.827957	54.245281	-7.582676
110	St. Lucia	NaN	70.754715	NaN
124	Mexico	36.021507	17.452829	-18.568678
141	Nicaragua	33.870968	7.547170	-26.323798
150	Panama	50.537636	28.773584	-21.764051
152	Peru	41.935482	22.169811	-19.765671
159	Paraguay	10.215054	15.094339	4.879286
172	El Salvador	21.505377	27.830189	6.324812
176	Suriname	61.290321	39.150944	-22.139378
190	Trinidad and Tobago	80.645164	40.566036	-40.079128
198	Uruguay	82.258064	91.981133	9.723068
199	United States	91.397850	82.547173	-8.850677
201	St. Vincent and the Grenadines	NaN	77.358490	NaN
202	Venezuela, RB	22.580645	1.886792	-20.693852

In [26]:

```
russ_data = df.loc[df['Country/Territory'] == 'Russian Federation']
russ_data = russ_data[['Country/Territory', 'Rank', 'Rank.23']]

russ_data.rename(columns={'Country/Territory': 'Country',
                          'Rank': '1996',
```

```
                                    'Rank.23': '2022'}, inplace=True)

russ_data['Rank_Difference'] = russ_data['2022'] - russ_data['1996']



russ_data
# rank_diff = pd.merge(rank_diff,df_by_region, on='Country', how='left')
# rank_diff
```

Out[26]:

	Country	1996	2022	Rank_Difference
163	Russian Federation	15.053763	19.339622	4.285859

In [27]:

```
rank_diff_and_rus =  pd.concat([rank_diff, russ_data], ignore_index=True)
rank_diff_and_rus
```

Out[27]:

	Country	1996	2022	Rank_Difference
0	Argentina	53.763439	36.320755	-17.442684
1	Bahamas, The	83.870964	84.433960	0.562996
2	Bolivia	25.268818	20.754717	-4.514101
3	Brazil	56.989246	32.075470	-24.913776
4	Barbados	90.860214	89.150940	-1.709274
5	Canada	96.236557	93.396225	-2.840332
6	Chile	90.322578	80.660378	-9.662201
7	Colombia	36.559139	41.037735	4.478596
8	Costa Rica	75.268814	66.981133	-8.287682
9	Cuba	63.440861	52.358490	-11.082371
10	Dominica	80.107529	69.339622	-10.767906
11	Dominican Republic	41.397850	37.264153	-4.133698
12	Ecuador	30.107527	29.716982	-0.390545
13	Grenada	80.107529	67.452827	-12.654701
14	Guatemala	23.655914	11.320755	-12.335159
15	Guyana	52.688171	45.283020	-7.405151
16	Honduras	14.516129	18.867924	4.351794
17	Haiti	9.139785	5.188679	-3.951106
18	Jamaica	61.827957	54.245281	-7.582676
19	St. Lucia	NaN	70.754715	NaN
20	Mexico	36.021507	17.452829	-18.568678
21	Nicaragua	33.870968	7.547170	-26.323798
22	Panama	50.537636	28.773584	-21.764051
23	Peru	41.935482	22.169811	-19.765671
24	Paraguay	10.215054	15.094339	4.879286
25	El Salvador	21.505377	27.830189	6.324812
26	Suriname	61.290321	39.150944	-22.139378
27	Trinidad and Tobago	80.645164	40.566036	-40.079128
28	Uruguay	82.258064	91.981133	9.723068
29	United States	91.397850	82.547173	-8.850677
30	St. Vincent and the Grenadines	NaN	77.358490	NaN
31	Venezuela, RB	22.580645	1.886792	-20.693852
32	Russian Federation	15.053763	19.339622	4.285859

In [28]:

```
# rank_diff = pd.merge(rank_diff_and_rus,df_by_region, on='Country', how='left')
# rank_diff
codes_df = df[['Country/Territory', 'Code']]
```

```
codes_df.columns = ['Country', 'Code']  # Переименуйте столбцы для соответствия
codes_df
```

Out[28]:

	Country	Code
0	Aruba	ABW
1	Andorra	ADO
2	Afghanistan	AFG
3	Angola	AGO
4	Anguilla	AIA
...	...	...
209	Serbia	SRB
210	South Africa	ZAF
211	Congo, Dem. Rep.	ZAR
212	Zambia	ZMB
213	Zimbabwe	ZWE

214 rows × 2 columns

In [29]:

```
rank_diff_and_rus = pd.merge(rank_diff_and_rus, codes_df, on='Country', how='left')
rank_diff_and_rus['Country'] = rank_diff_and_rus['Country'].replace('Russian Federa
rank_diff_and_rus
```

Out[29]:

	Country	1996	2022	Rank_Difference	Code
0	Argentina	53.763439	36.320755	-17.442684	ARG
1	Bahamas, The	83.870964	84.433960	0.562996	BHS
2	Bolivia	25.268818	20.754717	-4.514101	BOL
3	Brazil	56.989246	32.075470	-24.913776	BRA
4	Barbados	90.860214	89.150940	-1.709274	BRB
5	Canada	96.236557	93.396225	-2.840332	CAN
6	Chile	90.322578	80.660378	-9.662201	CHL
7	Colombia	36.559139	41.037735	4.478596	COL
8	Costa Rica	75.268814	66.981133	-8.287682	CRI
9	Cuba	63.440861	52.358490	-11.082371	CUB
10	Dominica	80.107529	69.339622	-10.767906	DMA
11	Dominican Republic	41.397850	37.264153	-4.133698	DOM
12	Ecuador	30.107527	29.716982	-0.390545	ECU
13	Grenada	80.107529	67.452827	-12.654701	GRD
14	Guatemala	23.655914	11.320755	-12.335159	GTM
15	Guyana	52.688171	45.283020	-7.405151	GUY
16	Honduras	14.516129	18.867924	4.351794	HND
17	Haiti	9.139785	5.188679	-3.951106	HTI
18	Jamaica	61.827957	54.245281	-7.582676	JAM
19	St. Lucia	NaN	70.754715	NaN	LCA
20	Mexico	36.021507	17.452829	-18.568678	MEX
21	Nicaragua	33.870968	7.547170	-26.323798	NIC
22	Panama	50.537636	28.773584	-21.764051	PAN
23	Peru	41.935482	22.169811	-19.765671	PER
24	Paraguay	10.215054	15.094339	4.879286	PRY
25	El Salvador	21.505377	27.830189	6.324812	SLV
26	Suriname	61.290321	39.150944	-22.139378	SUR
27	Trinidad and Tobago	80.645164	40.566036	-40.079128	TTO
28	Uruguay	82.258064	91.981133	9.723068	URY
29	United States	91.397850	82.547173	-8.850677	USA
30	St. Vincent and the Grenadines	NaN	77.358490	NaN	VCT
31	Venezuela, RB	22.580645	1.886792	-20.693852	VEN
32	Russia	15.053763	19.339622	4.285859	RUS

In [30]:

```
rank_diff_and_rus = rank_diff_and_rus.merge(df_by_region[['Code', 'Region']], on='C
rank_diff_and_rus
```

Out[30]:

	Country	1996	2022	Rank_Difference	Code	Region
0	Argentina	53.763439	36.320755	-17.442684	ARG	AME
1	Bahamas, The	83.870964	84.433960	0.562996	BHS	AME
2	Bolivia	25.268818	20.754717	-4.514101	BOL	AME
3	Brazil	56.989246	32.075470	-24.913776	BRA	AME
4	Barbados	90.860214	89.150940	-1.709274	BRB	AME
5	Canada	96.236557	93.396225	-2.840332	CAN	AME
6	Chile	90.322578	80.660378	-9.662201	CHL	AME
7	Colombia	36.559139	41.037735	4.478596	COL	AME
8	Costa Rica	75.268814	66.981133	-8.287682	CRI	AME
9	Cuba	63.440861	52.358490	-11.082371	CUB	AME
10	Dominica	80.107529	69.339622	-10.767906	DMA	AME
11	Dominican Republic	41.397850	37.264153	-4.133698	DOM	AME
12	Ecuador	30.107527	29.716982	-0.390545	ECU	AME
13	Grenada	80.107529	67.452827	-12.654701	GRD	AME
14	Guatemala	23.655914	11.320755	-12.335159	GTM	AME
15	Guyana	52.688171	45.283020	-7.405151	GUY	AME
16	Honduras	14.516129	18.867924	4.351794	HND	AME
17	Haiti	9.139785	5.188679	-3.951106	HTI	AME
18	Jamaica	61.827957	54.245281	-7.582676	JAM	AME
19	St. Lucia	NaN	70.754715	NaN	LCA	AME
20	Mexico	36.021507	17.452829	-18.568678	MEX	AME
21	Nicaragua	33.870968	7.547170	-26.323798	NIC	AME
22	Panama	50.537636	28.773584	-21.764051	PAN	AME
23	Peru	41.935482	22.169811	-19.765671	PER	AME
24	Paraguay	10.215054	15.094339	4.879286	PRY	AME
25	El Salvador	21.505377	27.830189	6.324812	SLV	AME
26	Suriname	61.290321	39.150944	-22.139378	SUR	AME
27	Trinidad and Tobago	80.645164	40.566036	-40.079128	TTO	AME
28	Uruguay	82.258064	91.981133	9.723068	URY	AME
29	United States	91.397850	82.547173	-8.850677	USA	AME
30	St. Vincent and the Grenadines	NaN	77.358490	NaN	VCT	AME
31	Venezuela, RB	22.580645	1.886792	-20.693852	VEN	AME
32	Russia	15.053763	19.339622	4.285859	RUS	ECA

Составление таблицы для региона Americas и РФ (WGI - rank)

In [31]:
```
result = pd.DataFrame({
    'Регион': ['-']*4,
    'Страна': ['-']*4,
    'WGI 1996': ['-']*4,
    'WGI 2022': ['-']*4,
    'Изменение (см. пункт 11)': ['-']*4
}, index=['mean_2022', 'max_2022', 'min_2022', 'Russia_2022'])
result
```

Out[31]:

	Регион	Страна	WGI 1996	WGI 2022	Изменение (см. пункт 11)
mean_2022	-	-	-	-	-
max_2022	-	-	-	-	-
min_2022	-	-	-	-	-
Russia_2022	-	-	-	-	-

In [32]:
```
AME_maxes.columns = ['Country', '2022']
AME_mins.columns = ['Country', '2022']


AME_maxes
result.at['mean_2022', 'Регион'] = "AME"
result.at['mean_2022', 'WGI 1996'] = AME_means['1996']
result.at['mean_2022', 'WGI 2022'] = AME_means['2022']
result.at['mean_2022', 'Изменение (см. пункт 11)'] = AME_means['2022'] - AME_means[

Max = rank_diff_and_rus[rank_diff_and_rus['Country'] == AME_maxes['Country'].values
result.at['max_2022', 'Регион'] = Max['Region'].values[0]
result.at['max_2022', 'Страна'] = Max['Country'].values[0]
result.at['max_2022', 'WGI 1996'] = Max['1996'].values[0]
result.at['max_2022', 'WGI 2022'] = Max['2022'].values[0]
result.at['max_2022', 'Изменение (см. пункт 11)'] = Max['Rank_Difference'].values[0



Min = rank_diff_and_rus[rank_diff_and_rus['Country'] == AME_mins['Country'].values[
result.at['min_2022', 'Регион'] = Min['Region'].values[0]
result.at['min_2022', 'Страна'] = Min['Country'].values[0]
result.at['min_2022', 'WGI 1996'] = Min['1996'].values[0]
result.at['min_2022', 'WGI 2022'] = Min['2022'].values[0]
result.at['min_2022', 'Изменение (см. пункт 11)'] = Min['Rank_Difference'].values[0

russ = rank_diff_and_rus[rank_diff_and_rus['Country'] == "Russia"]
result.at['Russia_2022', 'Регион'] = russ['Region'].values[0]
result.at['Russia_2022', 'Страна'] = russ['Country'].values[0]
result.at['Russia_2022', 'WGI 1996'] = russ['1996'].values[0]
result.at['Russia_2022', 'WGI 2022'] = russ['2022'].values[0]
result.at['Russia_2022', 'Изменение (см. пункт 11)'] = russ['Rank_Difference'].valu
result
```

Out[32]:

	Регион	Страна	WGI 1996	WGI 2022	Изменение (см. пункт 11)
mean_2022	AME	-	53.27957	45.59257	-7.686999
max_2022	AME	Canada	96.236557	93.396225	-2.840332
min_2022	AME	Venezuela, RB	22.580645	1.886792	-20.693852
Russia_2022	ECA	Russia	15.053763	19.339622	4.285859

12. Создание диаграммы размаха (boxplot) индекса WGI за 2022 для всех стран и для каждого региона в отдельности (estimate)

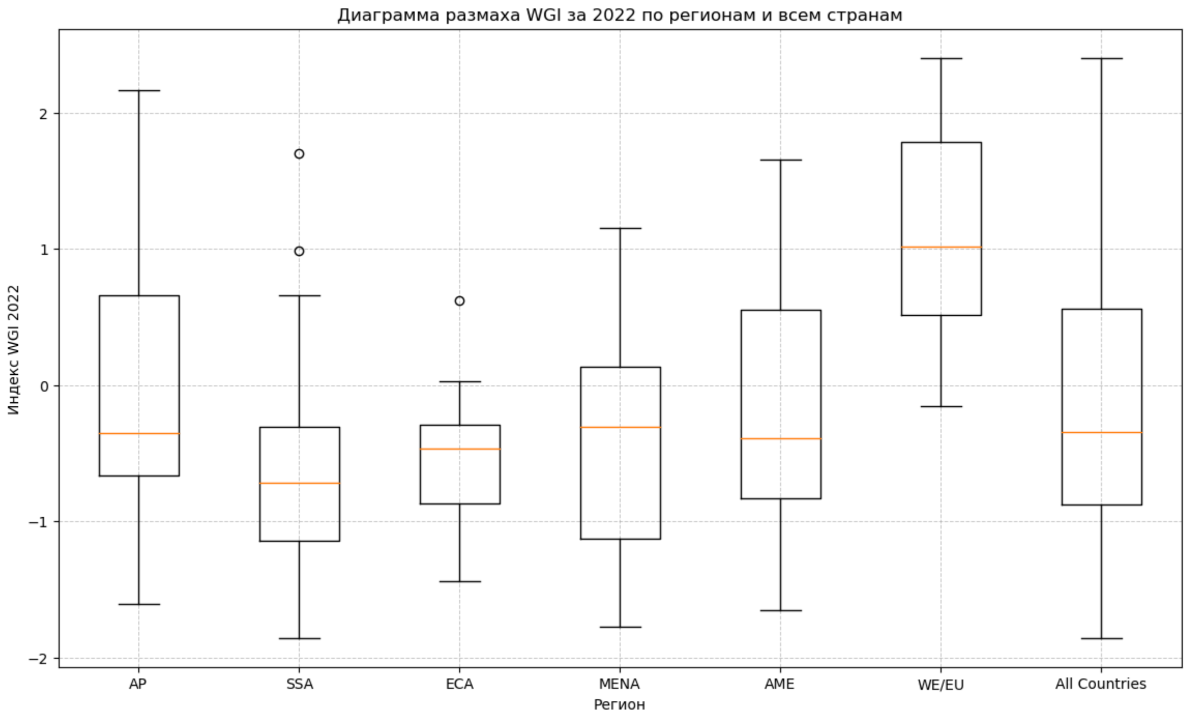
In [33]:
```
df_merged = pd.merge(df, df_by_region, on='Code', how='left')
df_merged = df_merged.rename(columns={'Estimate.23': 'WGI_2022'})
df_merged = df_merged[['Country', 'WGI_2022', 'Region']].dropna(subset=['WGI_2022',
df_merged
```

Out[33]:

	Country	WGI_2022	Region
2	Afghanistan	-1.183776	AP
3	Angola	-0.601941	SSA
5	Albania	-0.407876	ECA
7	United Arab Emirates	1.155336	MENA
8	Argentina	-0.447030	AME
...	...	...	...
208	Yemen	-1.679558	MENA
209	Serbia	-0.456188	ECA
210	South Africa	-0.319765	SSA
212	Zambia	-0.529200	SSA
213	Zimbabwe	-1.255139	SSA

177 rows × 3 columns

In [34]:
```
import pandas as pd
import matplotlib.pyplot as plt


df_merged = pd.merge(df, df_by_region, on='Code', how='left')
df_merged = df_merged.rename(columns={'Estimate.23': 'WGI_2022'})

# удаление  NaN
df_merged = df_merged[['Country', 'WGI_2022', 'Region']].dropna(subset=['WGI_2022',

regions = df_merged['Region'].unique()

data = [df_merged[df_merged['Region'] == region]['WGI_2022'] for region in regions]

# данные для всех стран
data.append(df_merged['WGI_2022'])

# метки для диаграммы размаха
labels = list(regions) + ['All Countries']

plt.figure(figsize=(14, 8))

plt.boxplot(data, labels=labels)

plt.title('Диаграмма размаха WGI за 2022 по регионам и всем странам')
plt.xlabel('Регион')
plt.ylabel('Индекс WGI 2022')
plt.grid(True, which='both', linestyle='--', linewidth=0.7, alpha=0.7)

plt.show()
```



Задача 2. Анализ рынка акций

In [35]:
```
import os
```

1. Загрузка данных в один dataframe из всех файлов в папке /data/stock.

In [36]:
```
folder_path = 'stock'

combined_df = pd.DataFrame()

for filename in os.listdir(folder_path):
    if filename.endswith('.csv'):
        file_path = os.path.join(folder_path, filename)

        df = pd.read_csv(file_path)

        stock_name = filename.replace('.csv', '')

        df = df[['Date', 'Close']]

        df.set_index('Date', inplace=True)

        df.rename(columns={'Close': stock_name}, inplace=True)

        if combined_df.empty:
            combined_df = df
        else:
            combined_df = combined_df.join(df, how='outer')

combined_df
```

Out[36]:

Date	AAPL	ABNB	ADBE	AMZN	CSCO	DBX	EBAY	GOOGL
2022-01-01	174.779999	153.970001	534.299988	149.573502	55.669998	24.750000	60.070000	135.303497
2022-02-01	165.119995	151.490005	467.679993	153.563004	55.770000	22.690001	54.590000	135.057007
2022-03-01	174.610001	171.759995	455.619995	162.997498	55.759998	23.250000	57.259998	139.067505
2022-04-01	157.649994	153.210007	395.950012	124.281502	48.980000	21.750000	51.919998	114.109497
2022-05-01	148.839996	120.870003	416.480011	120.209503	45.049999	20.840000	48.669998	113.762001
2022-06-01	136.720001	89.080002	366.059998	106.209999	42.639999	20.990000	41.669998	108.962997
2022-07-01	162.509995	110.980003	410.119995	134.949997	45.369999	22.740000	48.630001	116.320000
2022-08-01	157.220001	113.120003	373.440002	126.769997	44.720001	21.389999	44.130001	108.220001
2022-09-01	138.199997	105.040001	275.200012	113.000000	40.000000	20.719999	36.810001	95.650002
2022-10-01	153.339996	106.910004	318.500000	102.440002	45.430000	21.750000	39.840000	94.510002
2022-11-01	148.029999	102.139999	344.929993	96.540001	49.720001	23.559999	45.439999	100.989998
2022-12-01	129.929993	85.500000	336.529999	84.000000	47.639999	22.379999	41.470001	88.230003
2023-01-01	144.289993	111.110001	370.339996	103.129997	48.669998	23.230000	49.500000	98.839996
2023-02-01	147.410004	123.279999	323.950012	94.230003	48.419998	20.400000	45.900002	90.059998
2023-03-01	164.899994	124.400002	385.369995	103.290001	52.279999	21.620001	44.369999	103.730003
2023-04-01	169.679993	119.669998	377.559998	105.449997	47.250000	20.340000	46.430000	107.339996
2023-05-01	177.250000	109.769997	417.790009	120.580002	49.669998	23.020000	42.540001	122.870003
2023-06-01	193.970001	128.160004	488.989990	130.360001	51.740002	26.670000	44.689999	119.699997
2023-07-01	196.449997	152.190002	546.169983	133.679993	52.040001	26.950001	44.509998	132.720001
2023-08-01	187.869995	131.550003	559.340027	138.009995	57.349998	27.790001	44.779999	136.169998
2023-09-01	171.210007	137.210007	509.899994	127.120003	53.759998	27.230000	44.090000	130.860001
2023-10-01	170.770004	118.290001	532.059998	133.089996	52.130001	26.299999	39.230000	124.080002

	AAPL	ABNB	ADBE	AMZN	CSCO	DBX	EBAY	GOOG
Date								
2023-11-01	189.949997	126.339996	611.010010	146.089996	48.380001	28.180000	41.009998	132.52999
2023-12-01	192.529999	136.139999	596.599976	151.940002	50.520000	29.480000	43.619999	139.69000
2024-01-01	184.399994	144.139999	617.780029	155.199997	50.180000	31.680000	41.070000	140.10000
2024-02-01	180.750000	157.470001	560.280029	176.759995	48.369999	23.950001	47.279999	138.46000
2024-03-01	173.229996	166.669998	579.140015	175.389999	50.070000	23.840000	50.910000	138.50000
2024-03-12	173.229996	166.669998	579.140015	175.389999	50.070000	23.840000	50.910000	138.50000

28 rows × 25 columns

2. Получеине корряляционной матрицы для всех акций

In [37]:
```
correlation_matrix = combined_df.corr()
correlation_matrix
```

Out[37]:

	AAPL	ABNB	ADBE	AMZN	CSCO	DBX	EBAY	GOOGL	GTLB
AAPL	1.000000	0.617430	0.833129	0.665715	0.589552	0.740429	0.115591	0.806847	0.28237
ABNB	0.617430	1.000000	0.670509	0.830690	0.594365	0.332740	0.644140	0.780440	0.46060
ADBE	0.833129	0.670509	1.000000	0.819614	0.554172	0.816359	0.180354	0.915440	0.49655
AMZN	0.665715	0.830690	0.819614	1.000000	0.404820	0.478171	0.434078	0.912332	0.69064
CSCO	0.589552	0.594365	0.554172	0.404820	1.000000	0.496982	0.494938	0.600025	0.06885
DBX	0.740429	0.332740	0.816359	0.478171	0.496982	1.000000	-0.157363	0.669228	0.40251
EBAY	0.115591	0.644140	0.180354	0.434078	0.494938	-0.157363	1.000000	0.375794	0.25106
GOOGL	0.806847	0.780440	0.915440	0.912332	0.600025	0.669228	0.375794	1.000000	0.53547
GTLB	0.282373	0.460602	0.496556	0.690644	0.068856	0.402517	0.251066	0.535473	1.00000
HPQ	0.067074	0.390153	0.081518	0.235247	0.214262	-0.177013	0.744560	0.263251	0.09412
INTC	0.507251	0.738241	0.713875	0.816519	0.420854	0.390625	0.580047	0.826042	0.53544
META	0.705358	0.723419	0.873388	0.830910	0.374998	0.552874	0.190361	0.808784	0.46764
MSFT	0.790691	0.679204	0.913842	0.838702	0.391476	0.648164	0.127010	0.845993	0.45136
MU	0.606787	0.842928	0.817961	0.906932	0.472688	0.440043	0.512637	0.867191	0.54310
NFLX	0.701937	0.646901	0.821314	0.735466	0.497727	0.635239	0.138580	0.717756	0.45262
NVDA	0.633114	0.649664	0.802739	0.765294	0.320159	0.519374	0.087027	0.715287	0.40470
ORCL	0.769309	0.471504	0.785432	0.534556	0.463955	0.667833	-0.070414	0.618983	0.13857
PINS	0.640294	0.554616	0.804657	0.666996	0.384233	0.710191	-0.002757	0.640675	0.52545
SHOP	0.465147	0.696599	0.783919	0.824934	-0.144612	0.424923	0.338672	0.824313	0.85534
SPOT	0.687415	0.753797	0.863827	0.875779	0.424007	0.525305	0.296858	0.821587	0.54011
TCOM	0.439363	0.294269	0.533298	0.309545	0.257188	0.423136	-0.149330	0.322718	0.10361
TSLA	0.248385	0.353807	0.071508	0.302321	0.253808	0.037233	0.434899	0.326662	0.26090
TWLO	0.042914	0.429915	0.067604	0.314869	0.383777	-0.113102	0.753732	0.315410	0.31027
UBER	0.661323	0.680764	0.834611	0.796897	0.326346	0.595928	0.085736	0.737311	0.52139
XIACY	0.408747	0.564475	0.697612	0.654564	0.474311	0.382992	0.535223	0.680658	0.45366

25 rows × 25 columns

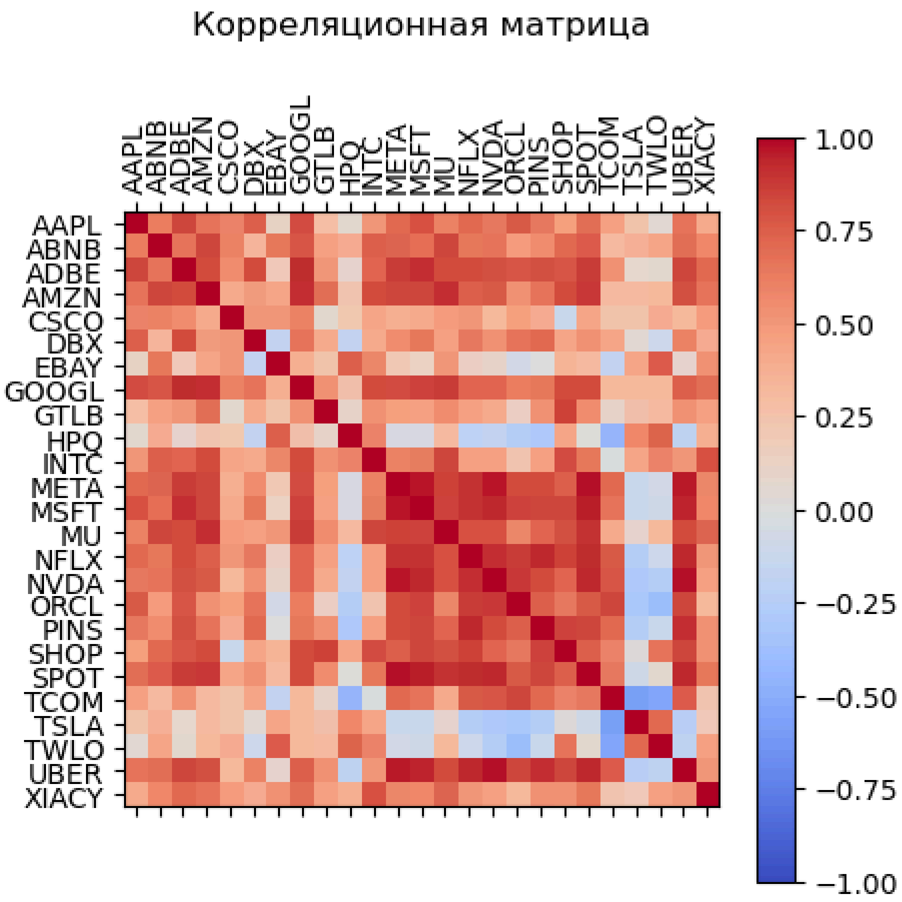
3. Отображение корреляционной матрицы в виде диаграммы

In [38]:

```
plt.figure(figsize=(16, 14), dpi=300)

# тепловая карта корреляционной матрицы
cax = plt.matshow(correlation_matrix, cmap='coolwarm', vmin=-1, vmax=1)
plt.colorbar(cax)

# добавление меток
plt.xticks(ticks=np.arange(len(correlation_matrix.columns)), labels=correlation_mat
plt.yticks(ticks=np.arange(len(correlation_matrix.columns)), labels=correlation_mat

plt.title('Корреляционная матрица', pad=20)
```

```
plt.show()
```

<Figure size 4800x4200 with 0 Axes>



4. Определение:\ Акции с максимальной положительной корреляцией (max)\ Акции с максимальной отрицательной корреляцией (min)\ Акции с минимальной корреляцией (которая больше всего соответствует отсутствию какой-либо корреляции (none)\ Для компании Netflix

In [39]:
```
df_NFLX = correlation_matrix.drop(["NFLX"], axis=1)
df_NFLX = df_NFLX.loc[["NFLX"]].T
df_NFLX.head()
```

Out[39]:

	NFLX
AAPL	0.701937
ABNB	0.646901
ADBE	0.821314
AMZN	0.735466
CSCO	0.497727

In [40]:
```
max_corr_id = df_NFLX.idxmax()[0]
df_NFLX.loc[[max_corr_id]]
```

Out[40]:

	NFLX
UBER	0.937042

In [41]:
```
min_corr_id = df_NFLX.idxmin()[0]
df_NFLX.loc[[min_corr_id]]
```

Out[41]:

	NFLX
TSLA	-0.251616

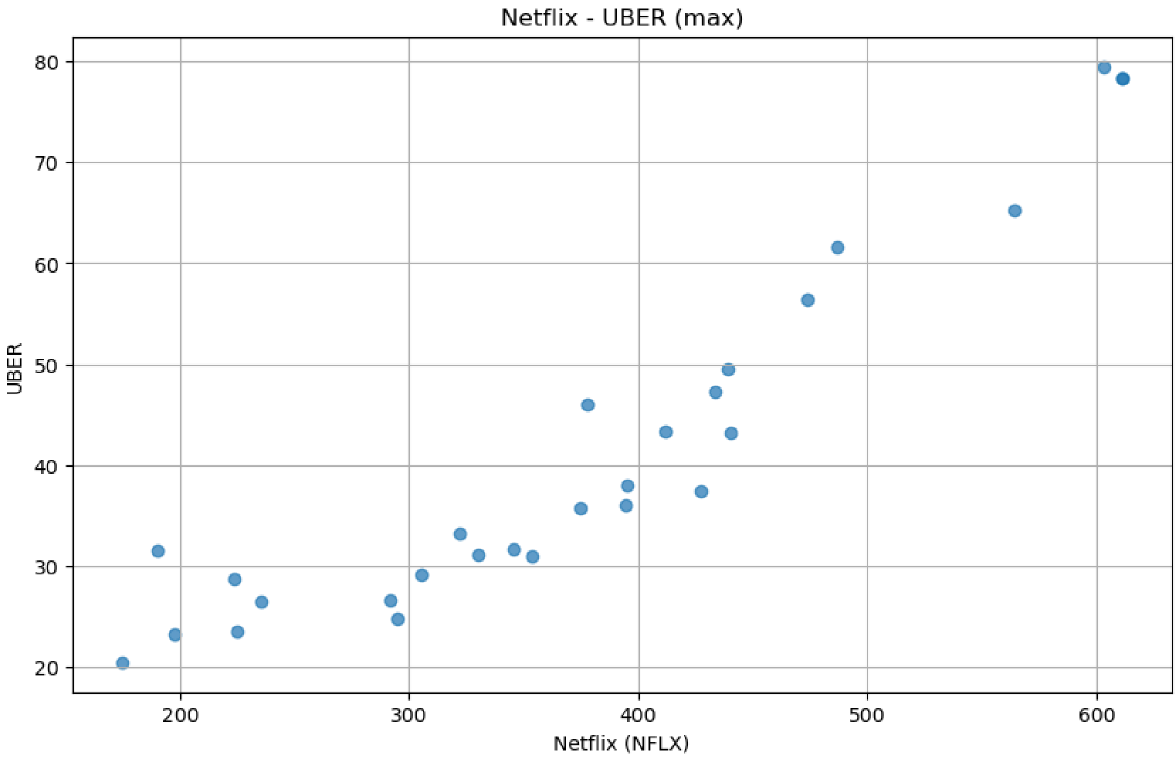
In [42]:
```
abs_min_corr = df_NFLX.fillna(0).abs().idxmin()[0]
df_NFLX.loc[[abs_min_corr]]
```

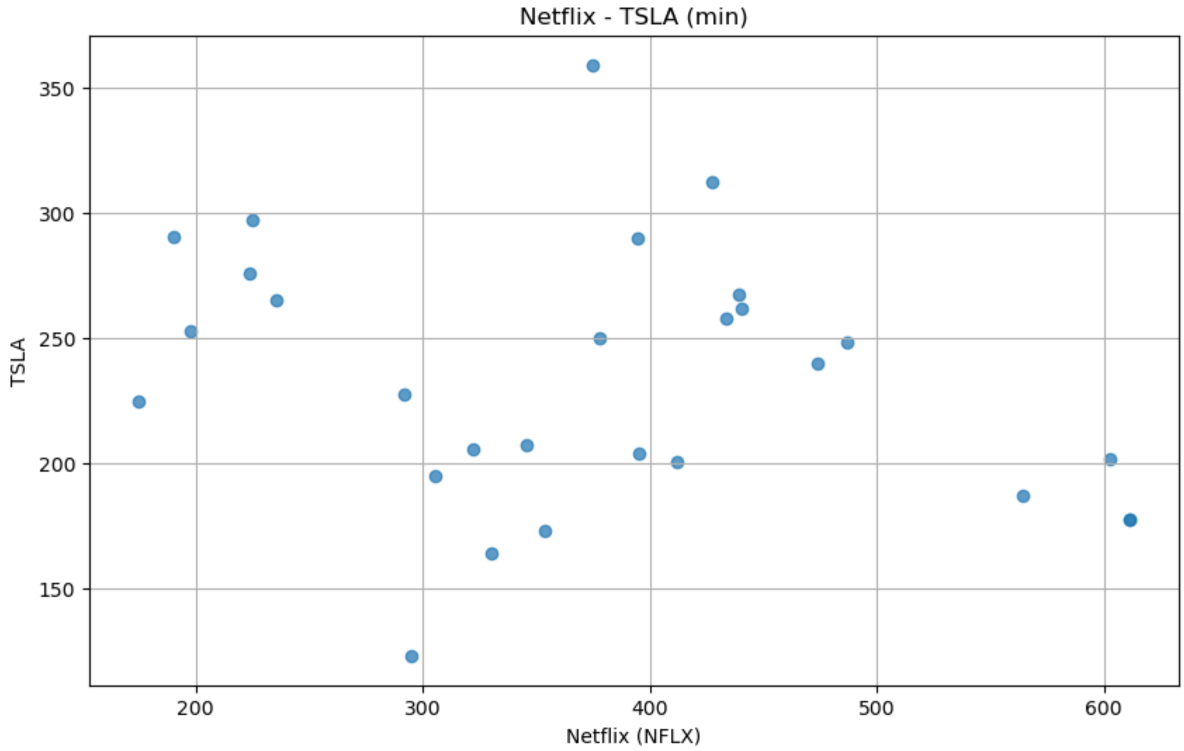
Out[42]:

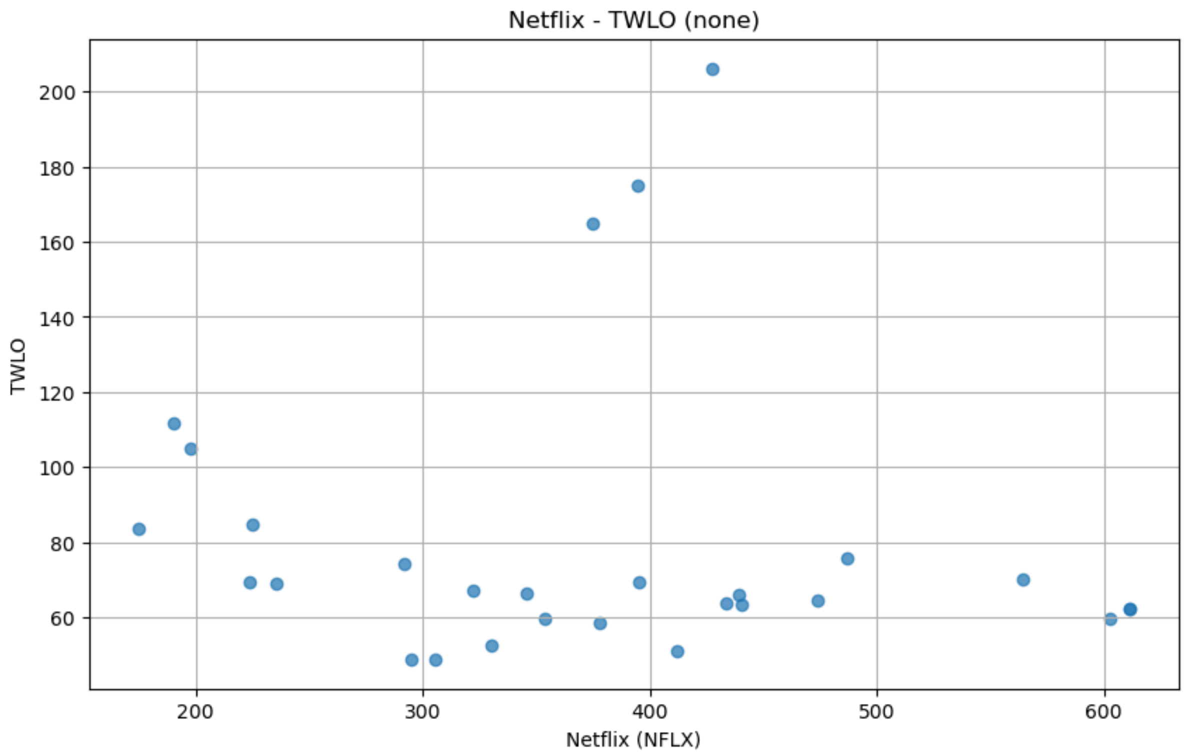
	NFLX
TWLO	-0.102302

5. Диаграммы разброса

In [43]:
```
def plot_scatter(x, y, x_label, y_label, title):
    plt.figure(figsize=(10, 6))
    plt.scatter(combined_df[x], combined_df[y], alpha=0.7)
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.title(title)
    plt.grid(True)
    plt.show()

plot_scatter('NFLX', max_corr_id, 'Netflix (NFLX)', f'{max_corr_id}', f'Netflix - {
plot_scatter('NFLX', min_corr_id, 'Netflix (NFLX)', f'{min_corr_id}', f'Netflix - {
plot_scatter('NFLX', abs_min_corr, 'Netflix (NFLX)', f'{abs_min_corr}', f'Netflix -
```







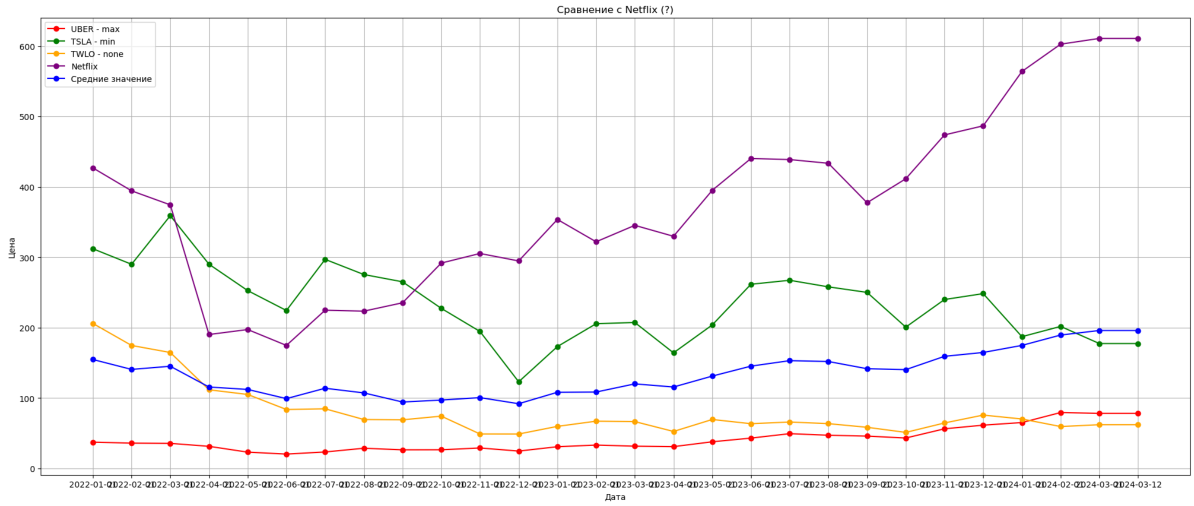
6. Средняя цена акций для каждого месяца

In [44]:
```
monthly_mean = combined_df.mean(axis=1)
monthly_mean
```

Out[44]:
```
Date
2022-01-01    154.857167
2022-02-01    140.774723
2022-03-01    145.272287
2022-04-01    115.763514
2022-05-01    112.316034
2022-06-01     99.256929
2022-07-01    114.014999
2022-08-01    107.380833
2022-09-01     94.437083
2022-10-01     97.227501
2022-11-01    100.671666
2022-12-01     92.028958
2023-01-01    108.279540
2023-02-01    108.613126
2023-03-01    120.210832
2023-04-01    115.778799
2023-05-01    131.258401
2023-06-01    145.426799
2023-07-01    153.207200
2023-08-01    152.016000
2023-09-01    141.760400
2023-10-01    140.454598
2023-11-01    159.367601
2023-12-01    164.859599
2024-01-01    174.886801
2024-02-01    189.609962
2024-03-01    196.083201
2024-03-12    196.083201
dtype: float64
```

7. Графики для акций из пункта 4 и средней из пункта 6

In [45]:
```
fig, ax = plt.subplots(figsize=(25, 10))

ax.plot(combined_df[max_corr_id], color='red', marker='o', label=f'{max_corr_id} -
ax.plot(combined_df[min_corr_id], color='green', marker='o', label=f'{min_corr_id}
ax.plot(combined_df[abs_min_corr], color='orange', marker='o', label=f'{abs_min_cor
ax.plot(combined_df['NFLX'], color='purple', marker='o', label='Netflix')
ax.plot(monthly_mean, color='blue', marker='o', label='Средние значение')

ax.set_xlabel("Дата")
ax.set_ylabel("Цена")

ax.legend()
ax.grid()

ax.set_title("Сравнение с Netflix (?)")

plt.show()
```



Вывод:

В результате выполнения работы был получен ценный опыт использования библиотек Python, таких как numpy, pandas и matplotlib для анализа данных на практических задачах.