Лабораторная работа №5

Извлечение данных с WEB-страниц. Пакет rvest.

5.1. Цель работы:

Научиться работать извлекать информацию cWEB-страниц с помощью инструментов языка R.

5.2. Общие сведения

Иногда возникает необходимость получить данные с веб-страниц и сохранить их в структурированном виде, для нашего курса это особенно актуально.

Инструменты веб-скрейпинга (web scraping) разрабатываются для извлечения данных с веб-сайтов. Эти инструменты бывают полезны тем, кто пытается получить данные из Интернета. Веб-скрейпинг — это технология, позволяющая получать данные без необходимости открывать множество страниц и заниматься копипастом. Эти инструменты позволяют вручную или автоматически извлекать новые или обновленные данные и сохранять их для последующего использования. Например, с помощью инструментов веб-скрейпинга можно извлекать наборы данных для дальнейшего анализа.

5.2.1. *Возможные сценарии использования инструментов веб-скрейпинга:*

- Сбор данных для маркетинговых исследований;
- Извлечение контактной информации (адреса электронной почты, телефоны и т.д.) с разных сайтов для создания собственных списков поставщиков, производителей или любых других лиц, представляющих интерес.
- Загрузка решений со StackOverflow (или других подобных сайтов с вопросами-ответами) для возможности оффлайн чтения или хранения данных с различных сайтов — тем самым снижается зависимость от доступа в Интернет.
- Поиск работы или вакансий.
- Отслеживание цен на товары в различных магазинах.

5.2.2. *Постановка проектной задачи*

Необходимо провести экспертный анализ документов, доступных только по запросам на веб портале и по относящимся к исходной задаче составить аналитический отчет. Звучит просто если бы не следующие нюансы:

1. БД документов недоступна напрямую.
2. Никакого API к порталу не существует.
3. Документы по прямой ссылке нельзя адресовать, только через механизм создания сессионного подключения и поиска.
4. Полный текст документа недоступен, его показ формируется JS скриптами для «постраничного» листания.
5. По каждому из подходящих документов необходимо, опираясь на plain text 1-ой страницы сделать реферативную сводку по его атрибутам.
6. Документов ~100 тыс, реально относящихся к задаче ~0.5%, времени на первый релиз ~ 1 неделя.

Копирование таблиц или списков с веб-сайтов – занятие~~нудное~~, ~~унылое,~~скучное и монотонное. К тому же этот процесс подвержен ошибкам и трудно воспроизводим. К счастью, существуют пакеты для Python и R, позволяющие автоматизировать выполнение подобных задач. В этой лабораторной мы изучим и опробуем библиотеку ***rvest.***

Решение задачи предполагает уверенное знание дерева DOM и CSS – селекторов.

Предварительно необходимо загрузить библиотеку:

```
install.packages("rvest")

library(rvest)
```

Для работы может также понадобится браузеры отладчик HTML-страниц, также может помочь плагин для Chrom: http://selectorgadget.com/ .

5.3. Краткий обзор функций библиотеки rvest

Наиболее важные функции в rvest:

- ***read_html()*** - Создание документа html из url, файла на диске или строки, содержащей html;
- Выделение фрагментов документа с помощью CSS-селекторов `html_nodes(doc, "table td")` (возможно, использование селекторов xpath с `html_nodes(doc, xpath = "//table//td")`).
- ЭИзвлечение компоненты с ***html_tag()*** (имя тега), ***html_text()*** (весь текст внутри тега), ***html_attr()*** (содержимое одного атрибута) и ***html_attrs()*** (все атрибуты).
- (Вы можете также использовать rvest с файлами XML распарсить в ***XML(),*** а затем извлекать компоненты с помощью ***xml_node(), xml_attr(), xml_attrs(), xml_text() и xml_tag()***.)
- Разбор таблиц в рамках данных с ***html_table().***
- Извлекать, изменять и отправлять формы с ***html_form(), set_values() и submit_form().***
- Обнаруживать и устранять проблемы с кодировкой ***guess_encoding()*** и ***repair_encoding()***.
- Навигация по сайту если вы в браузере с ***html_session(), jump_to(), follow_link()***, ***back(), forward(), submit_form()*** и так далее. Ссылки на документацию:

  https://www.rdocumentation.org/packages/rvest/versions/0.3.2
  https://cran.r-project.org/web/packages/rvest/rvest.pdf

5.4. Практические примеры

Пример 5.1:

Задача:
В качестве примера извлечем информацию о винодельческих и пивоваренных компаниях, расположенных в регионе Фингер-Лейкс (Finger Lakes), штат Нью-Йорк http://www.visitithaca.com/attractions/wineries .

На этой странице после заголовка расположены блоки, содержащие название компании и фото. Каждое фото также служит ссылкой на другую страницу, содержащую контактные данные соответствующей компании:

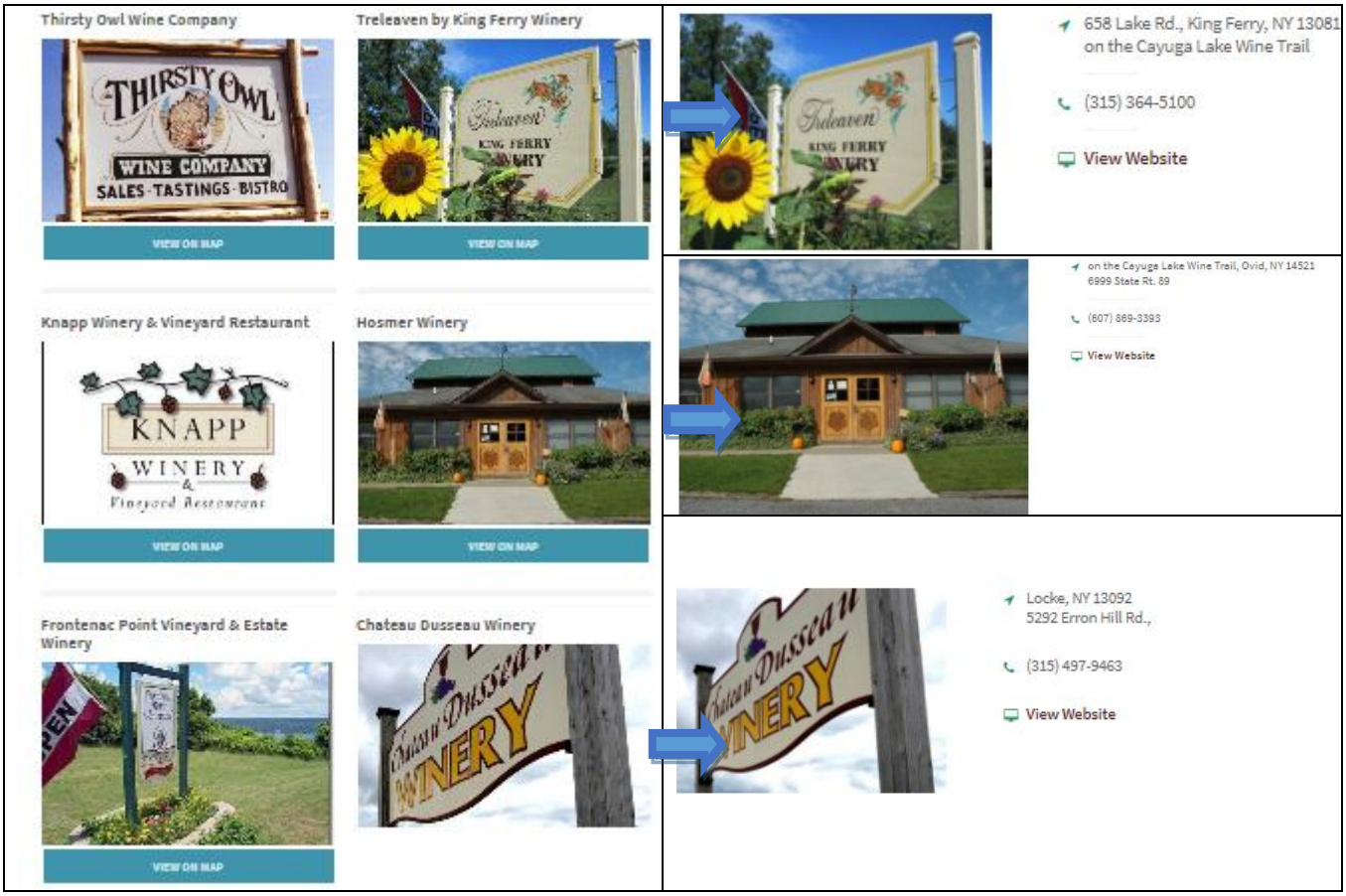


Рис.5.1. Постановка задачи поиска на странице. Структура элементов и переходы.

При сборе данных с веб-сайтов труднее всего определить, какие фрагменты HTML-кода необходимо извлечь. Обычно веб-страница представляет собой сложную структуру, состоящую из вложенных объектов, называемую объектной моделью документа (documentobject model, DOM). Соответственно, вы должны выяснить, какие фрагменты DOM вам нужны.

Чтобы сделать это, необходимо исследовать код веб-страницы с помощью инструментов разработчика, предоставляемых вашим браузером. Если вы используете Chrome или Firefox, открыть инструменты разработчика можно нажатием F12 (или Cmd + Opt + I для Mac), если вы используете Safari – Cmd + Opt + I.

Мы будем использовать Chrome. Обратите внимание, автор пакета Хэдли Уикхэм (Hadley Wickham) рекомендует использовать инструмент SelectorGadget, распространяемый в виде расширения Chrome, для поиска необходимых вам элементов веб-страниц. Также он рекомендует этот ресурс для изучения селекторов.

Чтобы следить за ходом изложения, откройте страницу со списком винодельческих компаний, которая используется в нашем примере.

Нажав [Ctrl+Shift+i] в Chrome, вы увидите нечто подобное тому, что изображено на рисунке ниже. Обратите особое внимание на инструмент для выбора элементов, обведенный красным цветом, и убедитесь в том, что у вас открыта вкладка Elements (элементы).

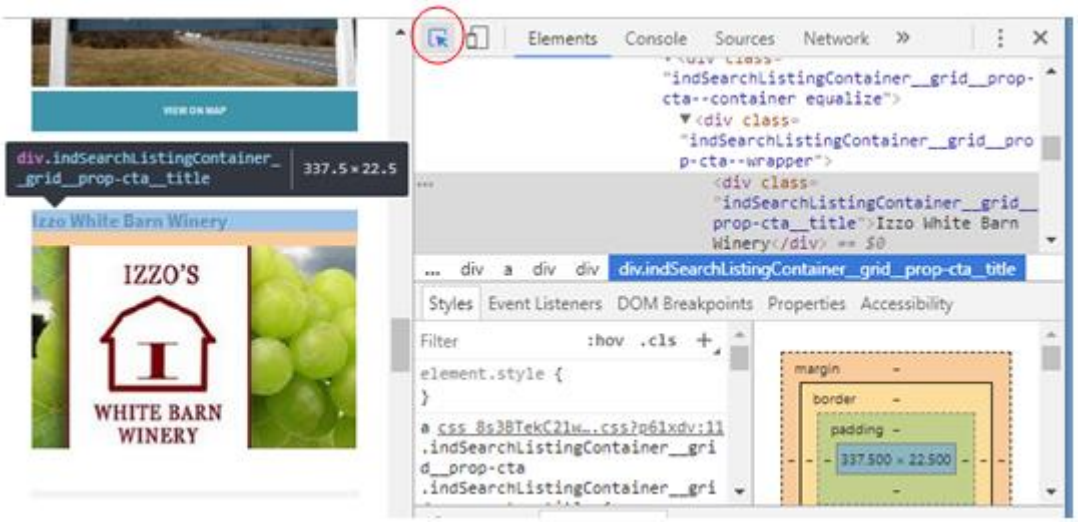


Рис.5.2. Выделение необходимого элемента и нахождение CSS – селектора для него.

Функция *html_nodes* извлекает весь узел из DOM, а затем функция *html_text* позволяет нам извлечь текст из узла. Обратите внимание на использование конвейера *%>%*, который передает результат выполнения функции *html_nodes* в функцию *html_text*.

```
# URL for the Visit Ithaca website, wineries page
> url<-read_html("http://www.visitithaca.com/attractions/wineries.html")
>
> # Pull out the names of the wineries and breweries
> selector_name<-"div.indSearchListingContainer__grid__prop-cta__title"
>
> fnames<-html_nodes(url, selector_name) %>% html_text()%>%as.array()

fnames
 [1] "Thirsty Owl Wine Company"
 [2] "Swedish Hill Vineyard"
 [3] "Frontenac Point Vineyard & Estate Winery"
 [4] "Izzo White Barn Winery"
 [5] "Finger Lakes Cider House"
 [6] "Cayuga Lake Wine Trail"
 [7] "Long Point Winery"
 [8] "Sheldrake Point Vineyard"
 [9] "Grisamore Cider Works"
[10] "Blackduck Cidery"
…
```

Извлекаем адреса фирм

Извлечь адреса фирм сложнее. Применив инструмент для выбора элементов, видим, что каждая фотография является ссылкой на страницу фирмы, и содержится в блоке:

```
<div class="indSearchListingContainer__grid attraction">
```

который, в свою очередь, содержит ссылку с классом **class="indSearchListingContainer__grid__prop-cta**:

```
<a data-org-href="/attractions/chateau-dusseau-winery"
href="/attractions/chateau-dusseau-winery" title=""
class="indSearchListingContainer__grid__prop-cta">
```

поэтому мы найдем все ссылки на соседние страницы, используя класс ссылки и, обращаясь к атрибуту найденного элемента:

```
selector_name<-".indSearchListingContainer__grid__prop-cta"
>
> fnames_addr<-html_nodes(url, selector_name) %>% html_attr("data-org-href")
>
> fnames_addr
 [1] "/attractions/thirsty-owl-wine-company"
 [2] "/attractions/americana-vineyards-crystal-lake-cafe"
 [3] "/attractions/blackduck-cidery"
 [4] "/meeting-facilities/treleaven-king-ferry-winery"
 [5] "/attractions/long-point-winery"
 [6] "/attractions/buttonwood-grove-winery"
 [7] "/attractions/swedish-hill-vineyard"
 [8] "/attractions/goose-watch-winery"
 [9] "/attractions/varick-winery-vineyard"
…
```

Однако, ко всем найденным ссылкам придется приклеить недостающую часть http://www.visitithaca.com с начала строки, функцией `paste0()` вот так:

```
fnames_addr1<-paste0("http://www.visitithaca.com",fnames_addr)
[1] "http://www.visitithaca.com/attractions/thirsty-owl-wine-company"
[2] "http://www.visitithaca.com/attractions/americana-vineyards-crystal-lake-cafe"
[3] "http://www.visitithaca.com/attractions/blackduck-cidery"
      …
```

Определим адрес первой компании:

```
url_sub<-read_html("http://www.visitithaca.com/attractions/thirsty-owl-wine-
company")
sub_selector_name<-".indMetaInfoWrapper"
> f1_address<-html_text(html_node(url_sub, sub_selector_name),trim=TRUE)
> f1_address
[1] "on the Cayuga Lake Wine Trail, Ovid, NY6861 Rt. 89,"
```

Если укажем, что нам нужны все узлы этого класса (`html_nodes()`), то получим и адрес, и телефон и строку, под которой находится ссылка на сайт.

```
f1_address<-html_text(html_nodes(url_sub, sub_selector_name),trim=TRUE)
> f1_address
[1] "on the Cayuga Lake Wine Trail, Ovid, NY6861 Rt. 89,"
[2] "(607) 869-5805"
[3] "View Website"
```

Если немного подумать, то можно извлечь адреса всех компаний.

Пример 5.2. Чтение данных из таблицы

Обратимся к ресурсу http://ladiesvenue.ru/olimpiada-2018-tablica-medalej-rezultaty-rossijskix-sportsmenov-15-fevralya .
На данной странице имеются две таблицы. Первые две строки скрипта (ниже) уже не требуют комментариев. В результате их работы будет сформирован список `nodes` из двух элементов (xml_nodeset), каждый их которых содержит таблицу.

```
> url= read_html('http://ladiesvenue.ru/olimpiada-2018-tablica-medalej-
rezultaty-rossijskix-sportsmenov-15-fevralya/')
> nodes = html_nodes(url, 'table'); nodes

> df1 = html_table(nodes[[1]])%>%as.data.frame()
> df2 = html_table(nodes[[2]])%>%as.data.frame()
```

Две последние строки скрипта указывают, что необходимо прочитать список `nodes`, как два отдельных элемента с помощью команды `html_table`, и при этом, преобразовать элемент списка в `data.frame`.

Получим: (фрагмент первой таблицы)

```
df1
   Место              Страна          Всего
1      1 Соединенные Штаты 13 15  8    36
2      2              Россия  8 10  6    24
3      3              Канада  8  4 16    28
4      4            Франция  7  8  5    20
5      5           Германия  7  8  4    19
6      6            Украина  7  7  8    22
7      7           Словакия  6  4  1    11
8      8           Беларусь  4  4  4    12 …
```

(фрагмент второй таблицы)

```
df2
                 М          Страна   З   С   Б Всего
1                1        НОРВЕГИЯ  14  14  11    39
2                2        ГЕРМАНИЯ  14  10   7    31
3                3          КАНАДА  11   8  10    29
4                4             США   9   8   6    23
5                5      НИДЕРЛАНДЫ   8   6   6    20
6                6          ШВЕЦИЯ   7   6   1    14
7                7     КОРЕЯ ЮЖНАЯ   5   8   4    17
8                8       ШВЕЙЦАРИЯ   5   6   4    15
9                9         ФРАНЦИЯ   5   4   6    15
10              10         АВСТРИЯ   5   3   6    14
11              11          ЯПОНИЯ   4   5   4    13
12              12          ИТАЛИЯ   3   2   5    10
13              13          РОССИЯ   2   6   9    17
…
31 Всего медалей  Всего медалей 103 102 102   307
```

Скриншот с сайта:

Олимпиада 2018: Таблица медалей. На каком месте Россия. Количество медалей у России 2018 таблица

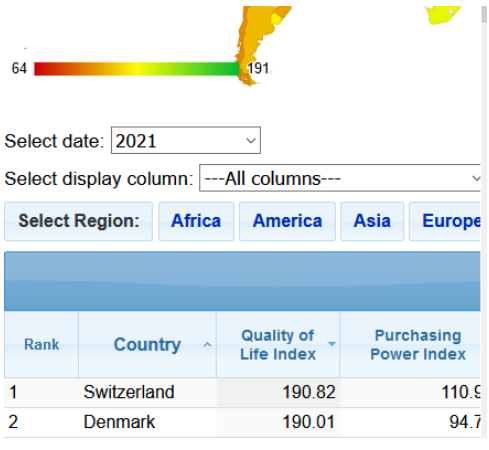
М	Страна	З	С	Б	Всего
1	НОРВЕГИЯ	14	14	11	39
2	ГЕРМАНИЯ	14	10	7	31
3	КАНАДА	11	8	10	29
4	США	9	8	6	23
5	НИДЕРЛАНДЫ	8	6	6	20
6	ШВЕЦИЯ	7	6	1	14
7	КОРЕЯ ЮЖНАЯ	5	8	4	17
8	ШВЕЙЦАРИЯ	5	6	4	15
9	ФРАНЦИЯ	5	4	6	15
10	АВСТРИЯ	5	3	6	14
11	ЯПОНИЯ	4	5	4	13
12	ИТАЛИЯ	3	2	5	10
13	РОССИЯ	2	6	9	17

5.5. Задания к лабораторной работе

1. В ходе лабораторной работы, необходимо собрать информацию об уровне жизни стран мира из таблиц сайта https://www.numbeo.com/quality-of-life/rankings_by_country.jsp?title=2021 с 2014 по 2021гг:

Для дополнительной информации по Рейтинг стран по уровню жизни можно использовать ссылку https://tyulyagin.ru/ratings/rejting-stran-mira-po-urovnyu-zhizni-2021.html

Это оценка общего качества жизни с использованием эмпирической формулы, которая учитывает:
— индекс покупательной способности (чем выше, тем лучше),
— индекс загрязнения (чем ниже, тем лучше),
— отношение цены на жилье к доходу (ниже). лучше),
— индекс прожиточного минимума (чем ниже, тем лучше),
— индекс безопасности (чем выше, тем лучше),
— индекс медицинского обслуживания (чем выше, тем лучше),
— индекс времени движения на дороге (чем ниже, тем лучше)
— климатический индекс (чем выше, тем лучше).



2. Каждый студент должен взять 5 стран (по варианту) .
3. Составить data.frame (возможно для каждой страны) так, чтобы иметь возможность **проанализировать с помощью графиков изменение рейтингов для всех 10 показателей для всех своих 5-ти стран, прокомменитровать в отчете результат.**

Необходимо нарисовать на одном и том же графике рейтинг всех 5 стран, проанализировать резутьтат. Проанализировать изменение во времени всех показателей указанных стран, подобрать наилучший (с вашей точки зрения) способ визуалазации.

4. С одной из страниц (https://kudago.com/spb/list/33-luchshih-muzeya-peterburga/ или https://tonkosti.ru/Музеи_Санкт-Петербурга) собрать информацию в data.frame, которя содержала бы: Название музея, его адрес и ссылку для перехода при клике на фото музея.

Варианты:

№	ФИО	Страны
1.	**Андреев Алексей Денисович**	Россия, Украина, Казахстан, Китай, Чили
2.	Воробьев Алексей Дмитриевич	Грузия, Мексика, Польша, Италия, Кипр
3.	Гринев Кирилл Владимирович	США, Португалия, Чехия, Хорватия, Россия
4.	Киреев Данил Максимович	Великобритания, США, Канада, Индия, Кения
5.	**Максюта Станислав Виталье**	Китай, Индия, Япония, Австралия, Австрия
6.	Малакеев Владимир Дмитриев	Франция, Куба, Аргентина, Болгария, Венгрия
7.	Миненков Владимир Владим	Турция, Греция, Египет, Австралия, Новая Зеландия
8.	Мишенькина Валерия Андреевн	Швеция, Болгария, Венгрия, Румыния, Нидерланды
9.	Павлоградская Мария Александровна – зам. старосты	Корея, Румыния, Чехия, Китай, Япония
10.	Парфинцов Егор Андреевич	Италия, Китай, Россия, Мексика, Куба
11.	Татарян Евгений Васильевич	Норвегия, Германия, Британия, Греция, Англия
12.	Фролова Анастасия Алексан	Швейцария, Швеция, Индонезия, Ливан, ОАЭ
13.	**Харисов Тимур Ринатович**	Бразилия, Индия, Ливан, Турция, Дания
14.	Шевякин Артём Анатольевич	Испания, Италия, Румыния, Греция, Дания
15.	Шемякин Никита Павлович	Куба, Аргентина, Болгария, Венгрия, Латвия
16.	Эзри Артём Александрович	Канада, США, Турция, Греция, Дания
17.	**Харисов Артур Марселевич**	Швейцария, Германия, Австралия, Латвия, Израиль
18.	Абрамов Иван Дмитриевич	Румыния, Чехия, Китай, Япония, Индия
19.	Бодяновская Анастасия Виталь	Нидерланды, Швейцария, Хорватия, ОАЭ, Египет
20.	Буткевич София Сергеевна	Россия, США, Украина, Азербайджан, Казахстан
21.	**Венедиктова Илона Сергеев**	Германия, Британия, Греция, Румыния, Норвегия
22.	**Гонтарев Александр Дмитр**	Россия, Украина, Белоруссия, Грузия, Армения.
23.	Колычев Егор Игоревич	Колумбия, Китай, Россия, Мексика, Бразилия
24.	Корнилов Кирилл Андреевич	Финляндия, Дания, Франция, Германия, Румыния
25.	Косян Артём Арменович	Канада, Австралия, Болгария, Польша, Украина
26.	Кузьменко Иван Сергеевич	Россия, Кипр, Израиль, Сербия, Куба
27.	Овдиенко Артём Сергеевич	Португалия, Чехия, Хорватия, Россия, США
28.	Диденко Тимур Алексеевич	Куба, Аргентина, Болгария, Венгрия, Латвия
29.	Полевая Полина Андреевна	Грузия, Мексика, Польша, Италия, Кипр
30.	**Попов Ивана Викторович**	Латвия, Эстония, Литва, Болгария, Венгрия
31.	Придава Александр Александ	Канада, США, Турция, Греция, Израиль
32.	**Романов Владислав Виталь**	Россия, Германия, Швеция, Франция, Финляндия
33.	**Тихонов Дмитрий Игоревич**	Португалия, Чехия, Хорватия, Россия, США

№	ФИО	Страна
1.	Ешев Нальбий Капланович	Финляндия, Греция, США, Германия, Англия
2.	Зайцев Александр Сергеевич	Греция, Швейцария, Корея, Вьетнам, Индия
3.	Иванова Виктория Алексеевна	США, Канада, Великобритания, Австрия, Дания
4.	Ищенко Владислав Владимирович	Великобритания, Мексика, Бразилия, Аргентина, Гонконг
5.	Кислица Данил Александров	Россия, Германия, Швеция, Франция, Финляндия

6.	Коваль Наталья Игоревна	Германия, Индонезия, Перу, Кения, Франция
7.	Малышев Денис Амилович	США, Китай, Россия, Казахстан, Белоруссия
8.	Миленченко Анастасия Романовна	Швейцария, Швеция, Индонезия, Ливан, ОАЭ
9.	Мингазетдинов Равиль Рустамович	Корея, Шри-Ланка, Индия, Пакистан, Греция
10.	Мищенко Никита Максимович	Италия, Турция, Испания, Румыния, Словения
11.	Москалец Ростислав Юрьевич	Россия, Австрия, Болгария, Польша, Украина
12.	Пинский Дмитрий Антонович	Германия, Британия, Греция, Румыния, Норвегия
13.	Прокопенко Евгений Константинович	Швейцария, Швеция, Индонезия, Ливан, ОАЭ
14.	Спиридонов Данил Александрович	Греция, Швейцария, Корея, Вьетнам, Индия
15.	Фролов Сергей Александрович	Россия, Германия, Швеция, Франция, Финляндия
16.	Шестак Виктория Александровна	Италия, Турция, Испания, Румыния, Словения
17.		Финляндия, Греция, США, Германия, Англия
18.		Германия, Британия, Греция, Китай, Россия,
19.		Швейцария, Германия, Австралия, Латвия, Израиль
20.		Германия, Австралия, Латвия, Израиль, Бразилия
21.		Египет, Австралия, Новая Зеландия, Турция, Греция
22.		США, Канада, Великобритания, Австрия, Дания
23.		Россия, Кипр, Израиль, Сербия, Куба