

Отчёт по индивидуальному проекту. Этап 6

Дисциплина: Операционные системы

Батова Ирина Сергеевна, НММбд-01-22

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	22

Список иллюстраций

3.1	Создание двух каталогов для английского и русского языков . . .	7
3.2	Редактирование файла 'languages.yaml'	8
3.3	Имя, фамилия, должность и университет	9
3.4	Интересы и образование	10
3.5	Описание владельца сайта	10
3.6	Текст поста о прошедшей неделе 1	10
3.7	Название поста о CD/CI	11
3.8	Текст поста о CD/CI	11
3.9	Навыки	11
3.10	Опыт	12
3.11	Достижения	13
3.12	Текст поста о прошедшей неделе 2	13
3.13	Название поста о языке разметки Markdown	14
3.14	Текст поста о языке разметки Markdown	14
3.15	Текст поста о прошедшей неделе 3	15
3.16	Название поста об оформлении отчета	15
3.17	Текст поста об оформлении отчета	15
3.18	Информация о первом проекте	16
3.19	Текст первого проекта	16
3.20	Информация о втором проекте	16
3.21	Текст второго проекта	17
3.22	Текст поста о прошедшей неделе 4	17
3.23	Название поста о языках научного программирования	17
3.24	Текст поста о языках научного программирования	18
3.25	Информация о посте о прошедшей неделе 5	18
3.26	Текст поста о прошедшей неделе 5	19
3.27	Информация о посте о шифровании	19
3.28	Текст поста о шифровании	20
3.29	Перевод названия поста о прошедшей неделе 5	20
3.30	Перевод текста поста о прошедшей неделе 5	20
3.31	Перевод названия поста о шифровании	21
3.32	Перевод текста поста о языках научного программирования . . .	21

Список таблиц

1 Цель работы

Целью шестого этапа индивидуального проекта является размещение двуязычного сайта на Github.

2 Задание

- Сделать поддержку английского и русского языков.
- Разместить элементы сайта на обоих языках.
- Разместить контент на обоих языках.
- Сделать пост по прошедшей неделе.
- Добавить пост на тему по выбору (на двух языках).

3 Выполнение лабораторной работы

1. Нам необходимо сделать на сайте поддержку английского и русского языков. Для этого в каталоге `~/work/blog/content` создаем два каталога: 'en' будет содержать все элементы сайта на английском языке, а 'ru' на русском (рис. 3.1).



Рис. 3.1: Создание двух каталогов для английского и русского языков

Далее в каталоге `~/work/blog/config/_default` открываем файл 'languages.yaml' и раскомментируем там необходимые строки так, чтобы у нас выполнялось поддержка двух языков на сайте (рис. 3.2).

```

1 # Languages
2 # Create a section for each of your site's languages.
3 # Documentation: https://wowchemy.com/docs/guide/language/
4
5 # Default language
6 en:
7   languageCode: en-us
8   # Uncomment for multi-lingual sites, and move English content into `en` sub-folder.
9   contentDir: content/en
10
11 # Uncomment the lines below to configure your website in a second language.
12 ru:
13   languageCode: zh-Hans
14   contentDir: content/ru
15 # title: Chinese website title...
16 # params:
17 #   description: Site description in Chinese...
18 # menu:
19 #   main:
20 #     - name: 传
21 #       url: '#about'
22 #       weight: 1

```

Рис. 3.2: Редактирование файла 'languages.yaml'

2. Далее нам необходимо перевести на английский язык все элементы сайта, а также посты и проекты, и внести соответствующие изменения во все необходимые места в каталоге 'en'. Я делала это соответственно каждому этапу индивидуального проекта.

Для начала открываем в каталоге ~/work/blog/content/en/authors/admin файл '_index.md'. Вводим на английском имя, фамилию, должность и название университета (рис. 3.3).


```

---
# Display name
title: Irina Batova

# Full name (for SEO)
first_name: Irina
last_name: Batova

# Status emoji
status:
  icon: 🍷

# Is this the primary user of the site?
superuser: true

# Role/position/tagline
role: Student of RUDN University

# Organizations/Affiliations to show in About widget
organizations:
  - name: RUDN University
    url: https://www.rudn.ru/

```

Рис. 3.3: Имя, фамилия, должность и университет

Вводим на английском свои интересы, короткое био и указываем образование (рис. 3.4).

```

# Short bio (displayed in user profile at end of posts)
bio: My interests in science are mathematics, programming, chemistry.

# Interests to show in About widget
interests:
- Mathematics
- Poetry
- Working with animals

# Education to show in About widget
education:
courses:
- School No. 57 (2010-2021)
- RUDN University (2022-2026)

```

Рис. 3.4: Интересы и образование

В конце файла вводим на английском более подробное описание владельца сайта (рис. 3.5).

```

I am Maloua Irina, a student of the RUDN University, Faculty of Science, direction "Mathematics and Mechanics". In addition to mathematics, I am also interested in literature and art.
(style="text-align: justify;")

```

Рис. 3.5: Описание владельца сайта

Далее переведем посты, которые мы делали на втором этапе индивидуального проекта. Для этого в каталоге `~/work/content/en/post` заходим в каталог `'lastweek'` и открываем файл `'index.md'`. Редактируем название, а в конец файла вводим текст поста на английском (рис. 3.6).

```

# Projects (optional).
# Associate this post with one or more of your projects.
# Simply enter your project's folder or file name without extension.
# E.g. 'projects = ["internal-project"]' references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects = []'.
projects: []

The past week has been completely normal. Passed two control - in linear algebra and mathematical analysis. In addition, I did lab work, history homework, and did some work in my spare time.

```

Рис. 3.6: Текст поста о прошедшей неделе 1

Далее в каталоге `'cdci'` открываем файл `'index.md'`. Аналогично редактируем название (рис. 3.7), в конец файла вводим текст поста (рис. 3.8).

```

title: "Continuous Integration and Continuous Deployment (CD/CI)"
subtitle: ""
summary: ""
authors: [admin]
tags: [Academic]
categories: [Demo]
date: 2023-03-17T16:50:31+03:00
lastmod: 2023-03-17T16:50:31+03:00
featured: false
draft: false

```

Рис. 3.7: Название поста о CD/CI

```

# Simply enter your project's folder or file name without extension.
# E.g. 'projects' = ["internal-project"] references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects' = [].
projects: []
---

In the context of modern development methods and DevOps, the abbreviations CI and CD are often used. CI stands for Continuous Integration and CD stands for Continuous Deployment (or Continuous Delivery).

Continuous Integration is a fundamental DevOps recommendation that developers regularly merge code changes into a central repository where automated builds and tests are performed. Developers who use continuous integration merge their changes back into the main branch whenever possible. Changes made by a developer are verified by creating an assembly and running automated tests on this assembly. When using continuous integration, a lot of attention is paid to test automation, as a result of which, when integrating new commits into the main branch, the application does not break.

There is also continuous delivery - the continuation of continuous integration, since it automatically deploys all code changes, that is, not only the testing process is automated, but also the product release process. Continuous deployment goes one step further than continuous delivery - every change that goes through the production pipeline is released to customers. No human intervention is required, and only an error during a test can prevent a new change from being deployed to production. Continuous deployment is used to speed up the customer feedback cycle.

Thus, continuous integration is part of both continuous delivery and continuous deployment. And continuous deployment is similar to continuous delivery, except that releases happen automatically.

The main disadvantage of using continuous integration and continuous deployment is that you have to write automated tests for each new feature, improvement or bug fix. In addition, you need a continuous integration server that can monitor the main repository and automatically run tests for each new committed commit. However, there are invaluable advantages - fewer bugs get into the production environment, because of this, the release build is much easier, testing costs are reduced, and the development process is accelerated.

```

Рис. 3.8: Текст поста о CD/CI

3. Переходим в каталог ‘~/work/blog/content/en’ и открываем файл ‘_index.md’.

Переводим информацию о навыках в блоке ‘features’, а также переводим название блока (рис. 3.9).

```

- block: features
  content:
    title: Skills
    items:
      - name: Copywriting
        description: 20%

      - name: Mathematical analysis
        description: 40%

      - name: Omnichannel promotion
        description: 60%

```

Рис. 3.9: Навыки

Переводим информацию об опыте в блоке ‘experience’, а также переводим название блока (рис. 3.10).

```
- block: experience
  content:
    title: Experience
    # Date format for experience
    # Refer to https://wowchemy.com/docs/customization/#date-format
    date_format: Jan 2006
    # Experiences.
    # Add/remove as many `experience` items below as you like.
    # Required fields are `title`, `company`, and `date_start`.
    # Leave `date_end` empty if it's your current employer.
    # Begin multi-line descriptions with YAML's `|2-` multi-line prefix.
    items:
      - title: Student
        company: RUDN University
        location: Moscow, Russia
        date_start: '2022-09-01'
        date_end: ''

      - title: Omnichannel content promotion
        company: Omnidocor
        location: Moscow, Russia
        date_start: '2022-04-15'
        date_end: ''
```

Рис. 3.10: Опыт

Переводим информацию о достижениях в блоке ‘accomplishments’, а также переводим название блока (рис. 3.11).

```

- block: accomplishments
  content:
    # Note: `&shy;` is used to add a 'soft' hyphen in a long heading.
    title: 'Accomplishments'
    subtitle:
    # Date format: https://wowchemy.com/docs/customization/#date-format
    date_format: Jan 2006
    # Accomplishments.
    # Add/remove as many `item` blocks below as you like.
    # `title`, `organization`, and `date_start` are the required parameters
    # Leave other parameters empty if not required.
    # Begin multi-line descriptions with YAML's `|2-` multi-line prefix.
    items:
      - certificate_url: https://mspo.msk.ru/
        date_end: '2020-06-01'
        date_start: '2020-03-01'
        description: ''
        organization: Moscow Student Pedagogical Team
        title: Young Counselor's School
        url: ''

```

Рис. 3.11: Достижения

Далее переведем посты, которые мы делали на третьем этапе индивидуального проекта. Для этого в каталоге `~/work/content/post/en` заходим в каталог `'lastweek2'` и открываем файл `'index.md'`. Редактируем название, а в конец файла вводим текст поста на английском (рис. 3.12).

```

projects: []
---
Last week I wrote a text in mathematical logic, and also did a lab in the subject "Operating systems" on the topic "Emacs text editor". In addition, we finally managed to plan a trip for the summer to friends abroad, we have already bought tickets.

```

Рис. 3.12: Текст поста о прошедшей неделе 2

Далее в каталоге `'markdown'` открываем файл `'index.md'`. Аналогично редактируем название (рис. 3.13), в конец файла вводим текст поста (рис. 3.14).

```

title: "Markdown language"
subtitle: ""
summary: ""
authors: [admin]
tags: [Academic]
categories: [Demo]
date: 2023-04-06T16:48:14+03:00
lastmod: 2023-04-06T16:48:14+03:00
featured: false
draft: false

```

Рис. 3.13: Название поста о языке разметки Markdown

```

# Simply enter your project's folder or file name without extension.
# E.g. 'projects = ["internal-project"]' references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects = []'.
projects: []
---

Markdown is a markup language used to create rich text using a text editor. It can be opened and modified in any text editor. Widely used for writing documentation and README files.

Markdown was created by writer and blogger John Gruber along with web activist Aaron Schwartz. They wanted to come up with the simplest and most convenient markup language with which text can be translated into a beautifully designed look. At the same time, the text must remain readable even without being reduced to a "beautiful" look. This is what Markdown turned out to be. It was created in 2004, but the language has gained the most popularity now. It is supported by a variety of blogging platforms, note-taking services, CMS, and even instant messengers.

Markdown contains basic elements that can be found in almost any README.md:
- first level heading for the title;
- highlighting in bold type important parts in the description;
- link with clear text;
- second-level heading for sub-items;
- a bulleted list to list the benefits.

In general, Markdown is used for quickly formatting an article for translation into PDF, layout of a text document for placement on the site, but without the use of HTML, writing documentation for the code, making notes in a personal digital notebook.

```

Рис. 3.14: Текст поста о языке разметки Markdown

4. На четвертом этапе мы добавляли к сайту ссылки на научные и библиометрические ресурсы, что не нуждается в переводе, поэтому сразу перейдем к переводу постов.

Для этого в каталоге `~/work/content/post/en` заходим в каталог `'lastweek3'` и открываем файл `'index.md'`. Редактируем название, а в конец файла вводим текст поста на английском (рис. 3.15).

```
# Projects (optional).
# Associate this post with one or more of your projects.
# Simply enter your project's folder or file name without extension.
# E.g. 'projects = ["internal-project"]' references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects = []'.
projects: []
---
```

Over the past week, I wrote a mathematical analysis test, completed 12 lab, and continued developing my site.

Рис. 3.15: Текст поста о прошедшей неделе 3

Далее в каталоге 'reportcreate' открываем файл 'index.md'. Аналогично редактируем название (рис. 3.16), в конец файла вводим текст поста (рис. 3.17).

```
title: "Report design"
subtitle: ""
summary: ""
authors: [admin]
tags: [Academic]
categories: [Demo]
date: 2023-04-24T13:09:14+03:00
lastmod: 2023-04-24T13:09:14+03:00
featured: false
draft: false
```

Рис. 3.16: Название поста об оформлении отчета

```
# Projects (optional).
# Associate this post with one or more of your projects.
# Simply enter your project's folder or file name without extension.
# E.g. 'projects = ["internal-project"]' references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects = []'.
projects: []
---
```

I use the Markdown markup language to design the report. At the beginning of the file, the number of the laboratory work on which the report is being made, the discipline and the full name of the student who completed the work is always indicated. The purpose of the work is indicated below, then each stage of working with screenshots is described step by step, a link is indicated for each screenshot. At the end of the work there is a conclusion consistent with the topic of the work, as well as answers to control questions.

The "*" sign is used to create headers. In addition, you can format the text - to set the bold style, you need to put double asterisks on both sides of the text, for italic style - single. To create unordered lists, asterisks or dashes are used (it is possible to nest one list in another), to quote the symbol >.

To generate a file from md to pdf and docs format, I use Pandoc.

Рис. 3.17: Текст поста об оформлении отчета

5. Переходим в каталог '~/work/blog/content/en/project/crpts' и открываем

файл 'index.md'. В начале файла переводим название проекта и краткое содержание (рис. 3.18).

```
---
title: Asymmetric Encryption
summary: Public key encryption methods - implementation scheme, advantages and disadvantages, the most famous systems
tags:
  - Deep Learning
date: '2023-05-05T00:00:00Z'
```

Рис. 3.18: Информация о первом проекте

В конец файла вводим переведенный текст проекта (рис. 3.19).

```
# slides (optional).
# Associate this project with Markdown slides.
# Simply enter your slide deck's filename without extension.
# E.g. 'slides = "example-slides"' references 'content/slides/example-slides.md'.
# Otherwise, set 'slides = ""'.
slides: example

---

In public-key encryption, each party creates a pair of keys, a public key and a private key. These keys are linked to each other, so a message encrypted with the public key can only be decrypted with the corresponding private key.

An important point is that when creating keys, it is possible to generate very large numbers in such a way that, knowing the public key, it is impossible to calculate the private key in any way. Thus, ensuring reliable encryption of messages is based on the knowledge of the private key exclusively by its owner. The idea of asymmetric encryption is based on a complex mathematical question that allows you to build one-way functions. Given  $x$ , the function  $y = f(x)$  is easily determined. But at the same time, it is very difficult to calculate  $x$  from a known value of  $y$ .  $y$  is used as the public key and  $x$  as the private key. For all encryption methods with a public key, the absence of other methods has not been mathematically strictly proven; the calculation of the private key by brute force. Cryptosystems with public keys differ in the form of one-way functions.

If it is necessary to transfer information from the interlocutor to the interlocutor, a single scheme for the implementation of asymmetric encryption operates. To begin with, the first interlocutor chooses an algorithm and a pair of public and private keys. The public key is sent to the second interlocutor over an open, insecure channel. After that, the second interlocutor encrypts the information using the sent public key and sends the received cipher back to the first interlocutor, who decrypts the message using the private key created at the beginning. At the same time, if it is necessary to establish a communication channel in both directions, then operations before decrypting messages are performed by both interlocutors. Thus, everyone will know their private, public keys and the public key of the interlocutor.

Thus, if we denote by  $E_A$  the public key of the interlocutor  $A$ , by  $D_A$  the private key of the interlocutor  $A$ , then  $y = E_A(u)$  is the result of encrypting the message  $u$ , and  $u = D_A(y)$  is the result of decrypting the message  $y$ . In this case, the transformations must be such that for any  $u$  the equality  $D_A(E_A(u)) = u$  holds true.
```

Рис. 3.19: Текст первого проекта

Далее переходим в каталог 'ports'. В начале файла переводим название проекта и краткое содержание (рис. 3.20).

```
---
title: Parallel and Serial Ports
summary: Features of using computer ports - connection points between a computer and peripheral devices
tags:
  - Deep Learning
date: '2023-05-05T00:00:00Z'
```

Рис. 3.20: Информация о втором проекте

В конец файла вводим переведенный текст проекта (рис. 3.21).


```

# Slides (optional).
# Associate this project with Markdown slides.
# Simply enter your slide deck's filename without extension.
# E.g. 'slides = "example-slides"' references 'content/slides/example-slides.md'.
# Otherwise, set 'slides = ""'.
slides: example
---

To communicate with peripheral devices, one or more I/O controller chips are connected to the computer bus. The main function of a port is to act as a connection point where a cable is connected from a peripheral device and to allow data to be transferred from one device to another. Usually, it is the very end of the connector located on the motherboard that is called the port. Any computer must have a minimum set of ports, without which operation is impossible - these are ports for connecting a monitor, mouse, keyboard, a network connection connector, a universal USB port and an audio connector for a sound card. If necessary, the number of computer ports can be increased using expansion boards connected to the main board.

Ports can be divided into two types - serial and parallel. They are divided depending on the type used for communication.

Serial ports (also called COM port) are ports through which peripheral devices are connected using a serial protocol. In other words, information is transmitted through them one bit at a time, bit by bit. The name "serial port" is assigned to a port that has the RS-232C standard, so although other interfaces (such as USB) also use a serial method of exchange, they are not classified as serial ports. Serial connectors typically have 9 to 25 pins, and a personal computer typically has one to four serial ports. Serial ports are usually built into the motherboard.

```

Рис. 3.21: Текст второго проекта

Далее переведем посты, которые мы делали на пятом этапе индивидуального проекта. Для этого в каталоге `~/work/content/en/post` заходим в каталог `'lastweek4'` и открываем файл `'index.md'`. Редактируем название, а в конец файла вводим текст поста на английском (рис. 3.22).

```

# Projects (optional).
# Associate this post with one or more of your projects.
# Simply enter your project's folder or file name without extension.
# E.g. 'projects = ["internal-project"]' references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects = []'.
projects: []
---

Due to the approach of the session, there was a lot of study and work. I continued to work on the site, completed the penultimate laboratory work.

```

Рис. 3.22: Текст поста о прошедшей неделе 4

Далее в каталоге `'languages'` открываем файл `'index.md'`. Аналогично редактируем название (рис. 3.23), в конец файла вводим текст поста (рис. 3.24).

```

title: "Scientific programming languages"
subtitle: ""
summary: ""
authors: [admin]
tags: [Academic]
categories: [Demo]
date: 2023-05-05T13:09:21+03:00
lastmod: 2023-05-05T13:09:21+03:00
featured: false
draft: false

```

Рис. 3.23: Название поста о языках научного программирования

```

# Projects (optional).
# Associate this post with one or more of your projects.
# Simply enter your project's folder or file name without extension.
# E.g. 'projects = ["internal-project"]' references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects = []'.
projects: []
---
As you know, there are many scientific programming languages in the world. In this post, we will talk a little about the most famous and used languages in various fields of science.

One important language is Bash, an updated Unix shell first released in the late 80s. Server configuration scripts are usually written in Bash. Applications: server administration, CLI tools, DevOps, virtualization.

COBOL belongs to one of the oldest but still used languages. It was created by Grace Hopper in the late 1950s and was intended to standardize the development of large-scale business applications on computers. An incredible 95% of ATM transactions go through COBOL systems, with many of these devices operating 24 hours a day, 365 days a year. Applications: banking/finance, international corporations, governments.

```

Рис. 3.24: Текст поста о языках научного программирования

6. Далее нам нужно написать пост о прошедшей недели на двух языках. Для создания поста о прошедшей неделе сначала переходим в каталог ‘~/work/blog/content/ru/post’ и создаем в нем каталог ‘lastweek5’. Открываем каталог и в нем открываем файл ‘index.md’.

Редактируем название и дату (рис. 3.25).

```

title: "Неделя с 15.05 по 21.05"
subtitle: ""
summary: ""
authors: [admin]
tags: [Academic]
categories: [Demo]
date: 2023-05-21T16:46:14+03:00
lastmod: 2023-05-21T16:46:14+03:00
featured: false
draft: false

```

Рис. 3.25: Информация о посте о прошедшей неделе 5

В конец файла вводим текст поста (рис. 3.26).

```
# Projects (optional).
# Associate this post with one or more of your projects.
# Simply enter your project's folder or file name without extension.
# E.g. `projects = ["internal-project"]` references `content/project/deep-learning/index.md`.
# Otherwise, set `projects = []`.
projects: []
---
```

Прошедшая неделя была обычной. Приближается сессия, поэтому много учебы.

Рис. 3.26: Текст поста о прошедшей неделе 5

Нам также нужно опубликовать пост на тему по выбору на двух языках. Я выбрала тему “Шифрование”. Для создания поста о шифровании сначала переходим в каталог ‘~/work/blog/content/ru/post’ и создаем в нем каталог ‘shifr’. Открываем каталог и в нем открываем файл ‘index.md’.

Редактируем название и дату (рис. 3.27).

```
title: "Шифрование"
subtitle: ""
summary: ""
authors: [admin]
tags: [Academic]
categories: [Demo]
date: 2023-05-21T13:09:14+03:00
lastmod: 2023-05-21T13:09:14+03:00
featured: false
draft: false
```

Рис. 3.27: Информация о посте о шифровании

В конец файла вводим текст поста (рис. 3.28).

```

# Projects (optional).
# Associate this post with one or more of your projects.
# Simply enter your project's folder or file name without extension.
# E.g. 'projects = ["internal-project"]' references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects = []'.
projects: []
---

Шифрование - это преобразование информации в целях ее сокрытия от неавторизованных лиц. Именно с помощью шифрования обеспечивается три состояния безопасности информации: конфиденциальность, целостность и идентифицируемость. В любом алгоритме шифрования важен ключ - он утверждает выбор конкретного преобразования. Ключ шифрования - набор цифр и букв, используемый алгоритмом для шифрования и расшифровки информации. При этом надежность ключа зависит от его длины в битах.

Шифрование бывает симметричным и асимметричным. Симметричное шифрование использует один и тот же ключ и для шифрования, и для расшифрования. А при асимметричном шифровании (шифровании с открытым ключом) ключи шифрования и расшифрования - разные, что делает этот вариант более надежным.

```

Рис. 3.28: Текст поста о шифровании

Теперь, аналогично предыдущим постам, нам нужно их перевести. Для этого в каталоге `~/work/content/en/post` создаем в каталог `'lastweek4'` и открываем файл `'index.md'`. Редактируем название (рис. 3.29), а в конец файла вводим текст поста на английском (рис. 3.30).

```

title: "15.05 - 21.05"
subtitle: ""
summary: ""
authors: [admin]
tags: [Academic]
categories: [Demo]
date: 2023-05-21T16:46:14+03:00
lastmod: 2023-05-21T16:46:14+03:00
featured: false
draft: false

```

Рис. 3.29: Перевод названия поста о прошедшей неделе 5

```

# Projects (optional).
# Associate this post with one or more of your projects.
# Simply enter your project's folder or file name without extension.
# E.g. 'projects = ["internal-project"]' references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects = []'.
projects: []
---

The past week has been normal. The session is coming up, so a lot of study.

```

Рис. 3.30: Перевод текста поста о прошедшей неделе 5

Далее создаем каталог 'shifr' и открываем в нем файл 'index.md'. Аналогично редактируем название (рис. 3.31), в конец файла вводим текст поста (рис. 3.32).

```
title: "Encryption"
subtitle: ""
summary: ""
authors: [admin]
tags: [Academic]
categories: [Demo]
date: 2023-05-21T13:09:14+03:00
lastmod: 2023-05-21T13:09:14+03:00
featured: false
draft: false
```

Рис. 3.31: Перевод названия поста о шифровании

```
# Projects (optional).
# Associate this post with one or more of your projects.
# Simply enter your project's folder or file name without extension.
# E.g. 'projects' = ["internal-project"] references 'content/project/deep-learning/index.md'.
# Otherwise, set 'projects' = [].
projects: []
---
```

Encryption is the transformation of information in order to hide it from unauthorized persons. It is with the help of encryption that three states of information security are provided: confidentiality, integrity and identifiability. In any encryption algorithm, the key is important - it approves the choice of a particular transformation. An encryption key is a set of numbers and letters used by an algorithm to encrypt and decrypt information. The strength of the key depends on its length in bits.

Encryption can be symmetrical or asymmetric. Symmetric encryption uses the same key for both encryption and decryption. And with asymmetric encryption (encryption with a public key), the encryption and decryption keys are different, which makes this option more reliable.

Рис. 3.32: Перевод текста поста о языках научного программирования

4 Выводы

В ходе шестого этапа индивидуального проекта я разместила двуязычный сайт на Github.