Peter the Great St. Petersburg Polytechnic University

—

Graduate school of cybersecurity

**GUIDELINES FOR LABORATORY WORK № 2**

**«Creating UML diagram»**

course «Object-Oriented Programming»

1. Performed by

student of gr. 5151004/10101 Veremeichuk Y.Y.

<signature>

1. Checked by

teacher Kubrin G.S.

<signature>

1. Saint-Petersburg
2. 2023

1 INTRODUCTION

* 1. PURPOSE

The main tasks of this laboratory work is studying the types of relationships between objects and classes, familiarization with the basic elements of defining, representing, designing and modeling software systems using the UML language, acquiring the skill of developing UML diagrams for application applications. And the second task is to rewrite the C code of some lab.

* 1. THEORY

The unified modeling language (UML) is a general-purpose visual modeling language that is intended to provide a standard way to visualize the design of a system.

UML provides a standard notation for many types of diagrams which can be roughly divided into 3 main groups: behavior diagrams, interaction diagrams, and structure diagrams.

Unified Modeling Language (UML) diagrams are a crucial tool in the field of software engineering and systems analysis. They serve as a standardized visual representation of software systems, enabling effective communication among stakeholders and facilitating the development, design, and maintenance of complex systems. Key reasons why UML diagrams are essential: visualization, communication, analysis, standardization.

2 MAIN BODY

2.1 Unit Cases

To begin with, the subject area was chosen – YouTube video hosting. Then entities capable of describing the subject area in general terms were identified.

Unit cases were generated for a detailed review of the work of YouTube.

Case 1: Viewing a Video on YouTube

Actors:

1. Registered user (User)
2. Unregistered user (Guest)

Steps:

1. The actor opens the YouTube website.
2. If the actor is not logged in, the system prompts to sign in to the account or register.
3. The actor searches for a video using the search bar.
4. The actor selects a video to view.
5. The video starts playing.
6. The actor controls playback (pause, play, and adjust quality).
7. The actor watches a summary of the video generated by Yandex Browser.
8. The actor requests a video translation provided by Yandex Browser
9. After viewing, the actor rates, leaves a comment, subscribes to the channel (for User).

Postconditions:

1. The actor can continue using YouTube.
2. He can exit the platform.

Exception:

If the video is unavailable, the system displays an error message.

Case 2: Uploading a Video to YouTube

Actor:

User

Steps:

1. The actor opens the YouTube website.
2. The actor selects the video he want to upload.
3. The system processes the video.
4. The system checks the implementation of hosting rules.

Postcondition:

The video is available for viewing on YouTube.

Exception:

If the video violates the platform's rules, the system reject the upload.

Case 3: Monetizing a YouTube Channel

Actors:

1. User
2. Channel

Steps:

1. The YouTube channel owner creates and fine-tunes their channel by regularly posting content.
2. The channel accumulates a sufficient number of subscribers and views.
3. The channel owner connects to the YouTube Partner Program.
4. YouTube Partner Program place ads on the channel`s videos.
5. After each ad-supported view, the channel owner earns money based on the number of views and user interactions with the ads (e.g., clicks or impressions).

Postcondition:

The channel owner generates income from monetizing their content on YouTube.

2.2 Classes and interfaces

Then, attributes and operations were designed for each entity in the form of a table according to UML notation, indicating data types and access modifiers.

Таблица 1 – Description of Guest

|  |  |
| --- | --- |
| Name | Guest |
| Entity type | Class |
| Note | Represents a person |
| Attributes |  |
| Operations | +LogIn(string login, string password): User  +SearchVideo(Search\_bar): Video  +StartVideo(): Video  +PauseVideo(): Video  +Configure(Video): Video  +ReadSummaryOfVideo(Video): Yandex\_Brouser  +GetVideoTranslation(Video): Yandex\_Brouser |

Таблица 2 – Description of User

|  |  |
| --- | --- |
| Name | User |
| Entity type | Class |
| Note | Represents a person registered in the system |
| Attributes | -name: string  -subscriptions: vector<Channel>  -link: string |
| Operations | +SubscribeChannel(Channel): void  +ReadSummaryVideo(Video): string  +GetVideoTranslation(Video): string  +RateVideo(Video, Rate): void  +WriteComment(Video): void  +AddVideo(Video): bool |

Таблица – Description ofSystem

|  |  |
| --- | --- |
| Name | System |
| Entity type | Class |
| Note | Represents system functional |
| Attributes |  |
| Operations | +CheckAuthorization(Guest): bool  +OfferLogIn(): void |

Таблица 2 – Description of Yandex Browser

|  |  |
| --- | --- |
| Name | Yandex\_Brouser |
| Entity type | Interface |
| Note | Equips the video hosting with user-friendly features |
| Attributes |  |
| Operations | +RetellVideo(Video): string  +TranslateVideo(Video): string |

Таблица – Description of Channel

|  |  |
| --- | --- |
| Name | Channel |
| Entity type | Class |
| Note | Working with User's videos |
| Attributes | -subscribers: vector<Guest>  -videos: vector<Video>  -bank: size\_t |
| Operations | +ConnectYTPP(): void  +ConfigAds(): void +UploadingVideo(): Video  +CheckVideo(Video): void  +BanVideo(Video): void |

Таблица – Description of Video

|  |  |
| --- | --- |
| Name | Video |
| Entity type | Class |
| Note | Resource |
| Attributes | -number\_of\_rates: Rate  -comments: vector<Comment>  -quality: size\_t  -link: string |
| Operations | +Play()  +Pause() |

Таблица 7 – Description of Search Bar

|  |  |
| --- | --- |
| Name | Search\_bar |
| Entity type | Interface |
| Note | A search bar with which you can find the content you are interested in |
| Attributes |  |
| Operations | +FindByTextQuery(): Interface\_VideoLib  +FindByVoiceQuery(): Interface\_VideoLib |

Таблица 8 – Description of Comment

|  |  |
| --- | --- |
| Name | Comment |
| Entity type | Class |
| Note | Some text user want to write under the video |
| Attributes | -comment: string  -rating: Rate |
| Operations | +Moderation(): void |

Таблица 9 – Description of Rate

|  |  |
| --- | --- |
| Name | Rate |
| Entity type | Class |
| Note | The rating that the user puts under the video or comment |
| Attributes | -finger\_up: size\_t  -finger\_down: size\_t |
| Operations | +AddFinger(finger\_up, finger\_down): void |

Таблица 10 – Description of YouTube\_Partner\_Program

|  |  |
| --- | --- |
| Name | YouTube\_Partner\_Program |
| Entity type | Interface |
| Note | Allows employers to advertise on channels |
| Attributes |  |
| Operations | +GetAds(): void  +AddAds(): void  +AssignPayout(Channel): void |

Таблица 11 – Description of Interface\_VideoLib

|  |  |
| --- | --- |
| Name | Interface\_VideoLib |
| Entity type | Interface |
| Note | Allows you to use the video library |
| Attributes |  |
| Operations | +OutputRecomVideos(): vector<Video> |

Таблица 12 – Description of Interface\_Video

|  |  |
| --- | --- |
| Name | Interface\_Video |
| Entity type | Interface |
| Note | Allows you to interact with video |
| Attributes |  |
| Operations | +RateVideo(): Rate  +CommentVideo(): Comment  +SubsribeChannel(): Channel  +TranslateVideo(): Yandex\_Brouser  +RetellVideo(): Yandex\_Brouser |

2.3 UML diagrams

Then a class diagram was constructed, including all the entities described above.

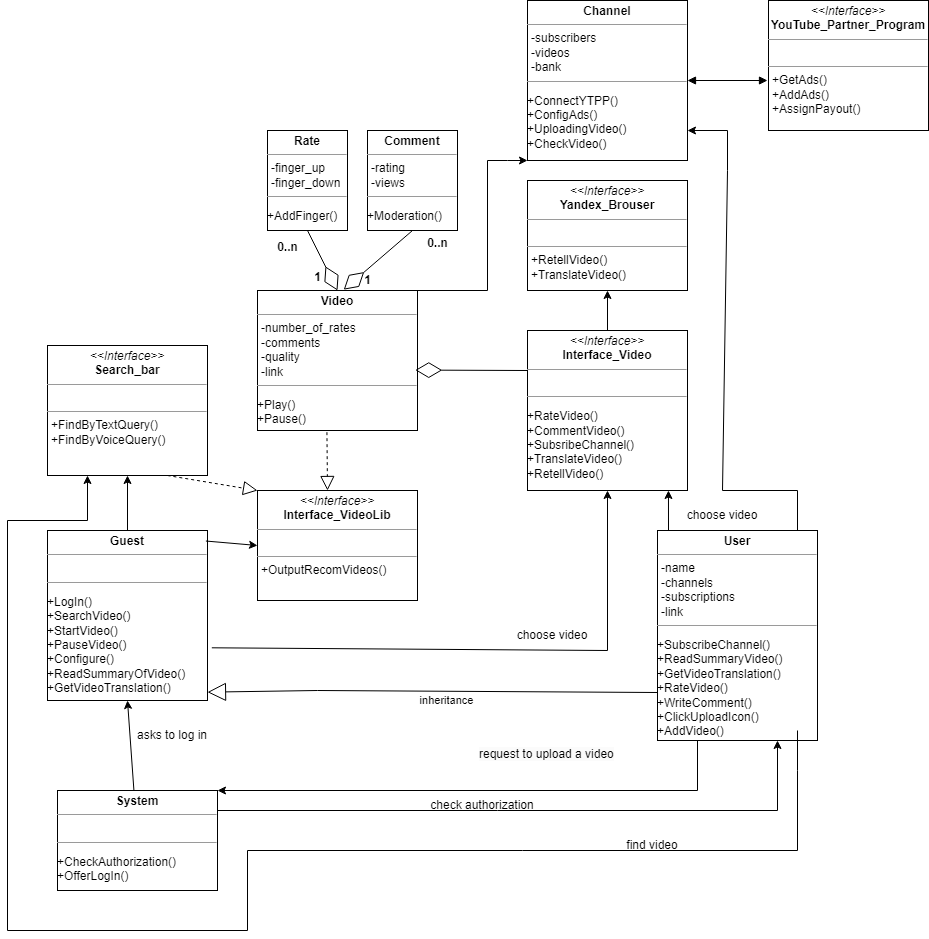


Рисунок – UML-диаграмма классов предметной области

After that, detailed UML diagrams were compiled for two unit cases: in the first case, a guest comes to Youtube and the system first checks whether he is authorized, and then offers him to do it. And in the second case, an authorized user wants to watch a video that opens using a special interface.



Рисунок 2 – Пример взаимодействия классов

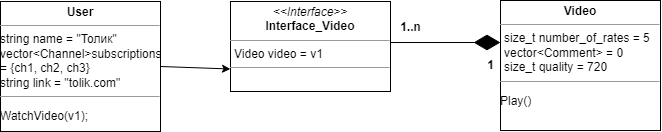


Рисунок 3 – Пример взаимодействия классов и интерфейса

As an additional task №4.1, a UML diagram of the objects of the subject area under consideration was created.

The object diagram in the UML modeling language is designed to demonstrate the totality of simulated objects and the relationships between them at a fixed point in time.

The object diagram describes specific instances of objects and directly correlates with the class diagram, which gives a general idea of the system configuration. It is used to document data structures and create static snapshots of object states taking into account real instances or prototypes.

For the subject area under consideration, a diagram was compiled reflecting the situation where the user Sveto4ka is subscribed to the user Vano31, who has one video that has a certain rating and comments.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 4 – UML-диаграмма объектов: два авторизованных пользователя

As an additional task № 4.2, UML activity diagrams were implemented.

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes, as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

Two situations were considered for the subject area under consideration. In the first, a user and an unauthorized guest log in to the system. The system checks whether the account is logged in, and if not, offers to do it. Guests and users get access to watch videos and watch one of them.

Таблица 13 – Description of Guest

|  |  |
| --- | --- |
| Name | Guest |
| Note | An unauthorized user enters the website. He has a choice: to log in or not. He can also find a video, select it and watch it. Or maybe immediately after logging in to the website, launch one of the suggested videos. |

Таблица 14 – Description of User

|  |  |
| --- | --- |
| Name | User |
| Note | An authorized user visits the website. He can find any video, select and watch it. Or maybe immediately after logging in to the website, launch one of the suggested videos. |

Таблица 15 – Description of System

|  |  |
| --- | --- |
| Name | System |
| Note | Some kind of entity that checks whether the user is authorized. If not, offers to do it. |

Таблица 16 – Description of SearchBar

|  |  |
| --- | --- |
| Name | SearchBar |
| Note | A search bar that accepts text or voice recording to search for a video. If the name of the video you are looking for matches the existing ones, the entity sends the found videos to the interface that outputs the recommended videos. |

Таблица 17 – Description of Interface\_VideoLib

|  |  |
| --- | --- |
| Name | Interface\_VideoLib |
| Note | The same interface that displays a list of recommended videos and gives the guest or user the opportunity to select a video. |

Таблица 18 – Description of Video

|  |  |
| --- | --- |
| Name | Video |
| Note | An entity that displays a real video as a material object. The video can be started and paused. |

Таблица 19 – Description of Interface\_Video

|  |  |
| --- | --- |
| Name | Interface\_Video |
| Note | A kind of shell over the Video. An entity that allows you to interact with the Video: leave comments, rate, subscribe to the channel, send a translation request and a summary of the video. |

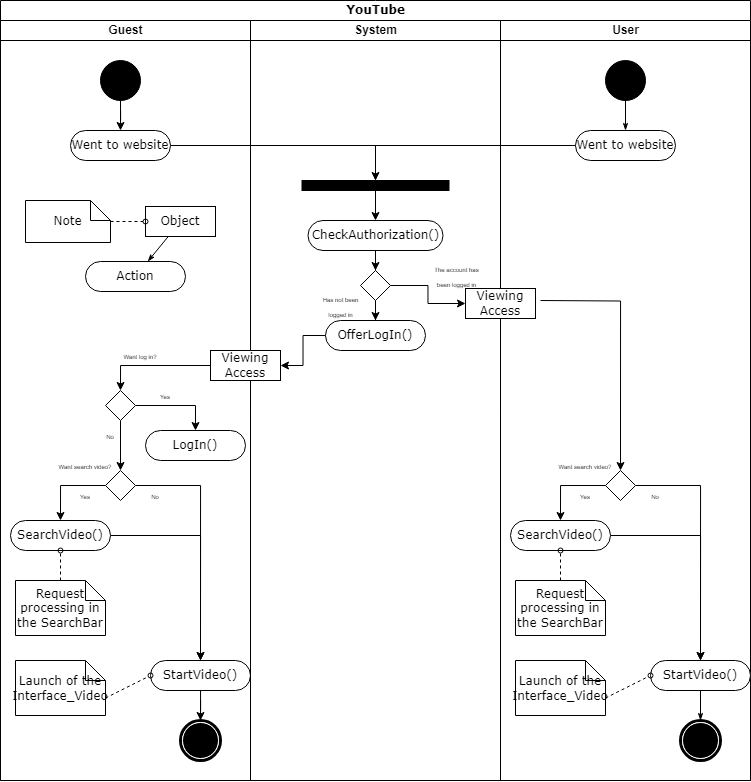


Рисунок 5 – UML диаграмма деятельности: вход пользователей и просмотр видео

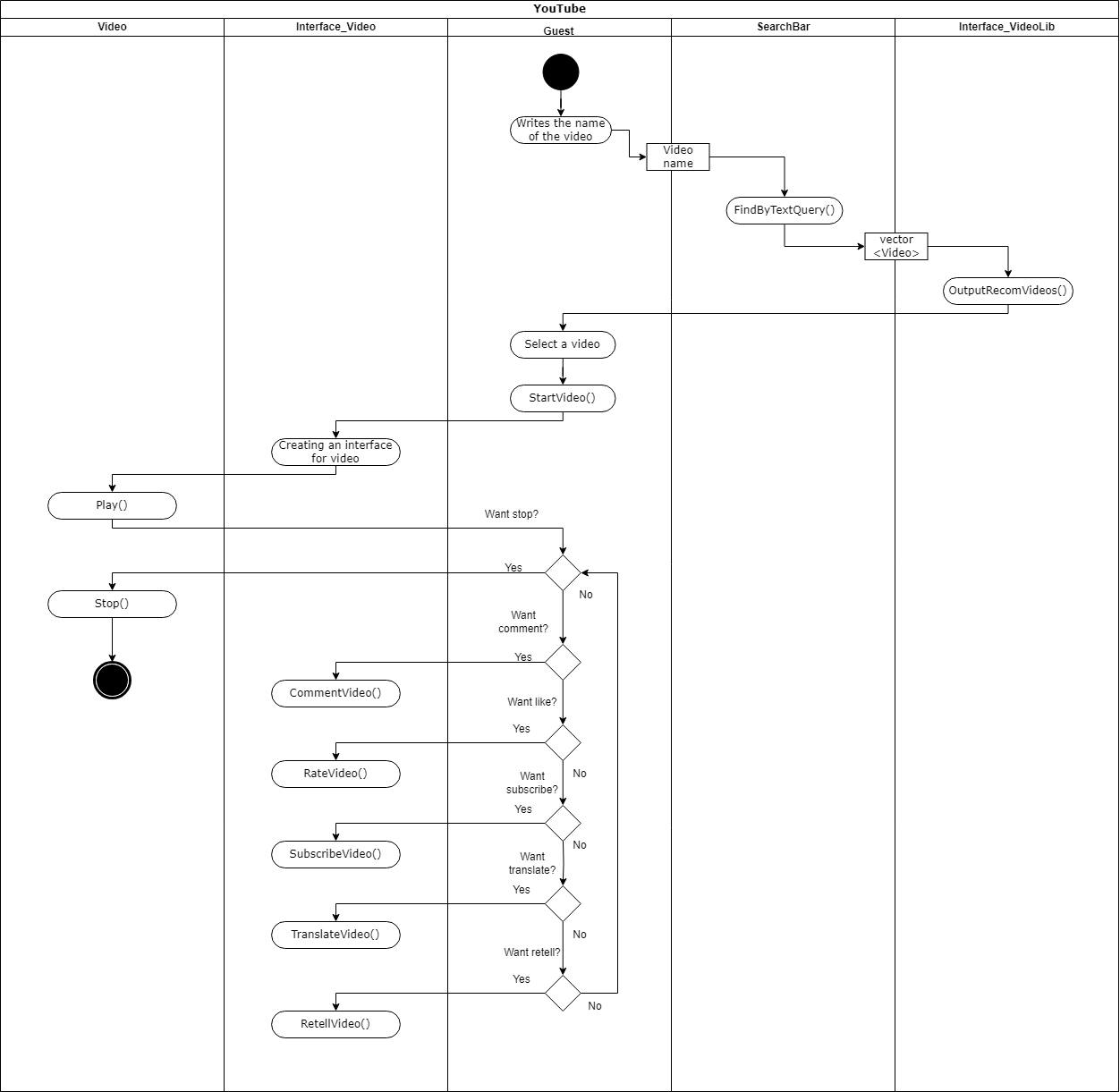


Рисунок 6 – UML диаграмма деятельности: поиск видео и взаимодействие с видео

2.4 Analyzing c-code and rewriting on c++

As the second part of the work, it was necessary to analyze the old c code and identify its shortcomings. The code "tic-tac-toe" was taken. The code you provided has several issues and potential problems.

1. Code duplication: There is a lot of code duplication.
2. Magic numbers: There are magic numbers throughout the code, such as the constants 1, 2, 3, 4, 5, and so on, which are used without explanation. It's a good practice to use named constants or provide comments to explain the significance of these numbers.
3. No error handling: The code lacks error handling for various scenarios, such as invalid user input in the humanMove function. It's important to handle errors gracefully and provide feedback to the user.
4. Exit on error: The code uses exit(-1) in some places to terminate the program when an error occurs. It's generally not a good practice to abruptly exit the program, as it doesn't provide a chance for cleanup or error reporting.

Then the game was rewritten in C++. All the problems described above have been solved. Code duplication was eliminated, magic numbers, global variables and emergency exits disappeared. The code has become more structured, concise and understandable.

A UML class diagram was written for the resulting code.

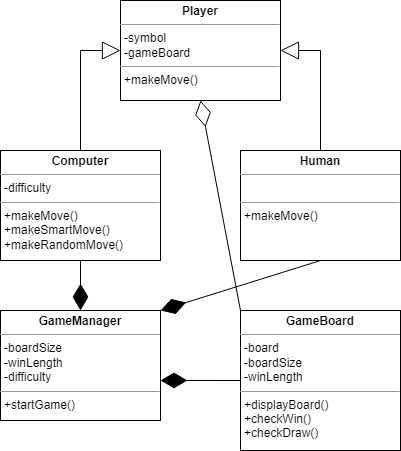


Рисунок 7 – UML-диаграмма для игры "Крестики-нолики на большом поле"

3 CONCLUSION

In the course of this laboratory work, the following goals were achieved: the study of the types of relationships between objects and classes in object-oriented programming. Familiarization with the basic elements of the definition, representation, design and modeling of software systems using the UML language. Acquisition of skills in developing UML diagrams for applications.

Additionally, as part of the laboratory work, the task of analyzing and rewriting the code for the game of tic-tac-toe was performed. This made it possible to apply the knowledge gained about the relationship between classes and objects in practice, as well as to improve the structure and readability of the code.

4 REFERENCES

1. Шилдт, Герберт. С++: базовый курс. : Пер. с англ. – Спб. : ООО “Диалектика”, 2019. – 624 стр. : ил. – Парал. тит. англ.
2. UML [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/UML> , свободный.