Peter the Great St. Petersburg Polytechnic University

—

Institute of cybersecurity and information security

**GUIDELINES FOR LABORATORY WORK № 2**

**« Creating UML diagram »**

course «Object-Oriented Programming»

Saint-Petersburg

2020

**Content**

[1. Task 3](#_Toc60585995)

[2. Theory 4](#_Toc60585996)

[3. Basic task 5](#_Toc60585997)

[3.1. UML creating 5](#_Toc60585998)

[3.2. Project refactoring 6](#_Toc60585999)

[4. Additional tasks 7](#_Toc60586000)

[4.1. Object diagram 7](#_Toc60586001)

[4.2 Additional diagram 7](#_Toc60586002)

[5. Report’s content 8](#_Toc60586003)

[6. Hints 9](#_Toc60586004)

1. Task

The main tasks of this laboratory work are:

* study the proposed theoretical material UML.docx;
* build a UML class diagram: specify access modifiers, as well as the power of class relationships, justify the type of relationship used;
* develop a UML class diagram for an existing project implemented in C;
* implement a program in C++ that reflects the developed class diagram and preserves the functionality of the project under study in C.

2. Theory

For the successful execution of the laboratory work it is necessary to study the following topics:

* UML.docx;
* importance of UML diagrams;
* code refactoring.

3. Basic task

3.1. UML creating

Choose a topic (see Table 1) or come up with your own and AGREE with the teacher. Describe the selected topic, show what features there are (screenshots with a description).

Table 1 — Task options

|  |  |  |
| --- | --- | --- |
| **№** | **Topic** | **Type of additional UML diagram** |
| 1 | Booking a flight “OneTwoTrip” | use case |
| 2 | Logistics of taxi “Uber” | component |
| 3 | The Poker app on mobile platforms | activity |
| 4 | Yandex-maps with monitoring public transport | sequence |
| 5 | Banking services in the mobile application “Sberbank” | state |
| 6 | Online store “Ozon” | component |
| 7 | Social network “Twitter” | activity |
| 8 | Tire service “Pit-stops” | sequence |

Analyzing the topic, determine which classes it can be divided into, and how these classes should be related, justify your point of view. After this stage, you will have a base for UML in text form — a justification for why you did so. Select entities (10-15) related to the specified subject area. For each entity, fill in the table (see Table 2). In the table, you need to design attributes and operations according to UML notation, specifying data types and access modifiers, and in the diagram itself, you can omit specifying the arguments and return values of operations.

Table 2 — Description of selected entity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Type** | **Role** | **Attributes** | **Operations** |
| Name | class/  interface/  … | brief description of the entity | class fields | class methods |
| Example | | | | |
| Guest | interface | Here is the interface to log in system. | - pass: int | + login(): bool |

Build a UML class diagram using 4 or more types of relationships. Specify access modifiers, as well as the power of class relationships.

Further on your designed system, make a use-case diagram and describe in detail (which functions are called in what order and with what arguments) any 3 use-cases from it.

3.2. Project refactoring

Choose your old project/coursework or take the open-source code. The main thing is that the project is in C and has at least 500 lines. Analyze the project: find bad solutions from the past, suboptimal solutions. Now do all the same work as in paragraph 3.1. only now your project is the topic.

After these steps, you need to REWRITE (and not just push functions into classes) the source code in C++. Describe whether you managed to fix all the problems described above. Describe how you solved problems and whether you managed to fix all them.

4. Additional tasks

Additional tasks should be done in order, otherwise points for them will not be counted.

4.1. Object diagram

You need to develop UML object diagram according to your chosen topic. You must show type of every object (actor, server, peer) and a way in which an object is made visible to other objects with which it interacts.

4.2 Additional diagram

You need to learn by yourself a UML notation of diagram shown in table 1 and build this UML diagram. Also you need to fill in the table describing every aspect shown in diagram. Table 3 and 4 show an example for use-case diagram and state diagram respectively.

Table 3 - Actors (Deal use-case diagram example)

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Custom | Actor, which selects item, initiates deal and pays for purchase |
| … | … |

Table 4 - States (Deal state diagram example)

|  |  |
| --- | --- |
| **State** | **State description** |
| Deal opened | Payment is received from custom and transferred to salesman. Exit from state is possible in case of incorrect payment or successful deal completion |
| … | … |

5. Report’s content

Your report must contain the following items, and MUST also be readable.

1. Goal.
2. Task.
3. Theory — you should briefly outline the theory you have learned to do your lab work.
4. Results — you should describe the process of work:
   1. designing UML:
      1. description of the selected subject area;
      2. analysis of the subject area;
      3. description of selected entity;
      4. UML diagram;
      5. Use-case diagram;
      6. Use-case descriptions.
   2. project refactoring:
      1. project description and problems analyzing;
      2. analysis of the project;
      3. UML diagram for C++ project;
      4. results C++ project.
5. Results of additional task completion:
   1. object diagram with shown types of objects;
   2. additional UML diagram;
   3. description table for additional diagram.
6. Conclusion.
7. References — a properly designed bibliographic list of references that you used.

6. Hints

There is no int A[5][2] declaration in C++, this is a forgotten C thing. Read about STL containers.

Mister Zavadskiy comment

Добрый день!

Продублирую информацию по второй лабораторной работе, которая в том или ином виде была озвучена на предыдущих парах.

Первая часть лабораторной – UML-диаграмма для выбранной вами предметной области. Число сущностей – 10-15. Помимо UML-диаграммы классов в данной части лабораторной необходимо сформулировать один use case в текстовом формате. Use case будет иметь примерно следующий вид:

Use case «purchase»

Actor: store’s customer

- “No problems” flow

1. Add one or more items into shopping cart

2. Select shipping options

3. Payout

- “Problem 1” flow

1. Add one or more items into shopping cart

2. Select shipping options (Shipping option doesn’t support bulky shipments)

3. Steps to resolve problem via additional user-system interactions steps… (e.g. suggest removing bulky items from shopping cart).

Я бы рекомендовал следующую последовательность выполнения первой части работы (после выбора темы): анализ предметной области с выделением ключевых сущностей, формулирование use case, проектирование системы с учётом поддержки сценария из use case.

Для выполнения второй части лабораторной рекомендую ориентироваться не столько на требование в 500 строк Си кода, сколько на возможность выделить в программе на Си (предметной области, задачу из которой решает программа) 4-5 взаимосвязанных сущностей. Один из хороших кандидатов для этой части работы - курсовой проект по программированию. В случае курсового, можно сосредоточиться на рефакторинге какого-то основного компонента (например, в случае если в курсовой работе реализовывалась игра - на механике игры, а, например, графическую подсистему оставить без изменений).