

**MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE**

**SIMON KUZNETS KHARKIV NATIONAL UNIVERSITY OF  
ECONOMICS**

# **OBJECT-ORIENTED PROGRAMMING**

**Guidelines**

**before completing a course project  
for advanced speciality 121**

**"Software Engineering" of the educational program**

**"Software engineering" of the first (bachelor's) level**

Authors: **Parfionov Yu.E.**

**Grizun L.E.**

**D.O. Bondarenko** is responsible for the release.

**Kharkiv  
HNEU  
2024**

UDC 004.415 (07)  
BBK 32.973-018r

**Stackers:** Parfyonov Yu.E.,  
Grizun L.E.

Approved at a meeting of the Department of Information Systems.  
Protocol No. 7 dated 12/21/2023

*Independent electronic text merezheve vidannya*

Methodical recommendations for writing a diploma project for students of higher education, specialty 121 "Software engineering" of the educational program "Software engineering" of the first (bachelor) level [Electronic resource]. Parfyonov Yu.E., Grizun L.E. – H.: View. KHNEU, 2024. - 40 p.

The issue of the organization of the diploma project is outlined, the requirements for the structure of the diploma project, methodical recommendations for the development of its structural elements are given.

It is recommended for applicants of the 121 "Software Engineering" specialty of the "Software Engineering" educational program of the first (bachelor's) level.

© Simon Kuznets Kharkiv National  
University of Economics, 2024

# Introduction

The current minds of the government rely on the economic management of the universal use of new information technologies. The wide possibilities of computerized methods for collecting, processing and types of necessary information from data will significantly promote the possibility of economical development, creating a more efficient process of economical processing Original decisions.

It is impossible to use the most advanced computerized technologies without software support. The importance of software fragmentation is increasing, as the trends in the development of computer technology indicate that on one side complexity and functional capabilities are rapidly increasing, and on the other side, This will require more sophisticated software to meet the needs of customers.

To create such software systems, leading experts can work with both the concepts of algorithmization and programming and various paradigms of programming, design, structural, functional and object-oriented. This allows them to disaggregate software systems from a wide variety of programming paradigms, since most current programming in this and other ways allows for the development of different paradigms.

A necessary element of successful mastery of the elementary material of the basic elementary discipline “Object-Oriented Programming” is independent work of students from technical literature, current languages, technologies and by means of programming. Increased effectiveness of course design.

The course project is an important stage in the preparation of students. The focus is on a greater understanding and consolidation of theoretical knowledge, acquired during the acquired discipline, and on the thorough development of practical skills in the development of software and necessary documentation.

These methodological recommendations establish uniform rules for the design, execution and completion of the course project. They are intended to assist students of specialty 121 “Software Security Engineering” with their course project.

# 1. Goal of course project

The method of course design is to consolidate and deepen knowledge from the discipline “Object-oriented programming”.

Working on a course project involves systematization, deepening and consolidation of knowledge that students have taken away during the course of training in this discipline. In the course of the course project, students develop skills in the practical application of removing knowledge from the ways of disassembling the application from the selection of current instrumental methods of development. In this case, the student is responsible for demonstrating his/her knowledge of specialized literature, government standards, references and other information technology materials.

At the hour of completing the course project, the student can show knowledge of:

- subject area depending on the assignment;

- basic concepts of algorithmization and programming;

- object-oriented approach to programming;

- there will be immediate prospects for the development of programming technologies;

- current instrumental features used for the development of software systems.

A healthcare professional may be able to:

- finalize the task setting and analysis;

- decompose the software system into subsystems;

- dismantle the hidden architecture of the software system;

- detail the underlying architecture of the software system;

- expand the program implementation of the singing programming system (similarly to the work program of the initial discipline “Object-oriented programming”);

- Vikorize standard libraries of language programming or software platforms;

- document the output code of the program;

- vikoristovat features of fragmentation of programs and extraction of pre-development information;

- disassemble and complete the necessary documentation.

Working on a course project in the first place signifies the theoretical and special preparation of the student and, as a result, prepares him before graduation for a more complex and final stage in the initial process - thesis project. The student may view the work on the course project as a kind of “rehearsal” for the graduation project.

The results of training and competence, which forms the illumination of the component, are shown in table. 1.

Table 1

**The results of training and competence, which form the course project**

Navchannya results	Competencies for which he is responsible rejuvenate your health and improve your health
PH8	SK1, SK10, SK13
RN13	SK2, SK3

de, PH7. Know and practice the fundamental concepts, paradigms and basic operating principles of the logical , instrumental and computational aspects of software engineering.

PH8. Consider breaking down the human-machine interface.

RN13. Know and develop methods for developing algorithms, designing software and data structures and knowledge.

SK1. It is necessary to identify, classify and formulate possibilities for software security.

SK2. Reality is involved in the design of software, including modeling (formal description) of its structure, behavior and functioning processes.

CK3. The importance of developing architecture, modules and components of software systems.

SK10. The value of accumulating, accumulating and systematizing professional knowledge in the creation and maintenance of software and the recognition of importance in the course of all life.

SK13. It is necessary to acquire and master the tools from the development and maintenance of software.

## 2. Organization of course project

The clear implementation of the course project requires a clear organization of work from the moment the project is selected and completed.

Coursework projects are supported by contributions from the Department of Information Systems. The course project guide recommends the student the main literature, guiding him to the breakdown of necessary project solutions.

A student may be assigned a topic for a course project based on a variety of recommended topics (div. supplement A). He is also entitled to the right to independently choose those for the project. If a topic is introduced by a student, it may be discussed and agreed upon with the course project.

After confirmation of the choice, students are assigned to the course project. The assignment contains the topic of the course project, the output data for the project, the replacement of the explanatory note, the beginning and completion of work on the course project, the schedule of the initial process, the schedule of the final stages of the course project. The supervision of the course project was given to supplementary teacher B.

After studying the literary elements, the developer puts forward the final plan for the course project and discusses it with the kerivnik. During the negotiation process, the output data for the design line is specified , which regulates the work of developers on the project. After this, the student refines the plan for working on the project, and then proceeds to design. During the completion of the course project, the student is responsible for regularly consulting with the kernel, submitting work materials for revision in accordance with the schedule for completing the stages of the course project.

Materials for the final course project: explanatory note in manual and electronic form, output code of the software installation, installer program, presentation of the proof in electronic form, the student may submit a copy for verification Ishe designated line.

All course projects are organized by the Department of Information Systems according to a confirmed schedule.

At the end of the day, the student briefly informs about the assignment, the project decisions that were accepted, the results obtained, and confirms the exam. The testimony is accompanied by a demonstration of an electronic presentation, which may contain, at a minimum, the following slides:

1. Title slide with final data for the course project: topic for the project, P.I.B. Wiconavian, P.I.B. Kerivnik Toscho.
2. Place of presentation.
3. Relevance of topics and meta course project.
4. A short production of the plant.
5. Mathematical description of the problem (number of slides).
6. UML diagram of classes that implement the basic business logic of the software system.
7. Vikoristan tools and technologies.
8. Visnovki for the results of the course project.
9. Final slide.

### **3. Structure and outline of the course project**

The course project consists of an explanatory note and other materials, including a program block, which is broken down according to the assignment.

The length of the explanatory note is approximately 20 – 30 side pages in A4 format (without supplements). The materials of the explanatory note appear to be bound.

In the table 1 shows the structure of the explanatory note for the course project.

### **4. Methodical recommendations before breaking down the structural elements of the course project 's explanatory note**

The title arch is the first side of the explanatory note. Vin Place the data that is given in this order:

news about Vikonavian robots;

full name of the document;

signatures of other personalities, including kerivnik and robots;

rik folding explanatory note;

The butt of the titular arkush was pointed at supplementary V.

An abstract is a short statement instead of an explanatory note in order to contain the main factual facts and conclusions necessary for a basic understanding of it.

The abstract is intended to be concise, informative and contain information that allows you to make decisions about the extent of reading the explanatory note.

Table 1

**Structure of an explanatory note for a course project**

Structural elements of an explanatory note	Number of pages
Title arkush	1
Head of course project	2 (on one arch)
Essay	1
Zmist	1
Enter	1
1 Project specification	3 - 6
1.1 Setting up a task	12
1.2 Vimogi before software security	12
1.3 Mathematical description of the problem	12
2 Software documentation	10 - 17
2.1 Software system architecture	13
2.2 Testing the software system	3 - 5
2.3 Rozlartya of the software product	12
2.4 Kerivnitstvo koristuvach	4 - 7
2.4.1 Software product assignment	1
2.4.2 Vikorista software product	2 - 4
2.4.3 Congratulations to Koristuvacheva	12
Visnovki	1
List of Wikorista Gerels	1
Add-ons	

Abstract guilty of revenge:

abstract text;

overflow of key words.

The text of the abstract must contain information provided in the explanatory note in the following sequence:

object of investigation or disaggregation;  
meta robots;  
research methods and equipment;  
the results are new;  
main technological technical and operational characteristics and displays;  
interactions with other robots;  
recommendations for improving the results of the course project;  
galuz zastosuvannya;  
the significance of work and innovation;  
forecasts about the development of the object of investigation or fragmentation.

The abstract must be compiled in no more than 500 words, and it is important that it fits on one side in A4 format.

Keywords are used to describe the essence of the project and to provide general information about the development. They should be placed after the text of the abstract. The flow of key words is mixed from 5 to 15 words (words), directed by great writers in a naming form in a row through the coma.

An example of the abstract was provided by appendix G.

To the point include: entry; consistently list the names of all sections, subsections and points of the main part of the explanatory note ; visnovki; overflow posilan; names of addendums and page numbers to indicate the origin of the material.

An example of the explanatory note was provided to appendix D.

At the beginning, it is important to note the importance of information systems in modern minds, the role of information systems in the subject area that is seen in the course project, the role of data saving methods in information systems, and the objectives of the course project. , a summary of the extensive program (noting that it allows you to automate , technologies and language programming, which were used by vikorstans, with the disaggregated programs), the final part.

The first section "Specification of the project" contains the formulation of the task, making it possible to provide a software and mathematical description of the problem.

Subsection 1.1 "Setting the task" is aimed at setting the task for breaking down the software product according to the version that will be produced. A

description of the setting of the task for any topic, as a minimum, may contain information similar to that provided in the supplement by J.

Subsection 1.2 “Software support” contains functional and non-functional capabilities for the software system.

Functional systems clearly describe what the software system must do and how the input data must be processed.

Non-functional features represent the power of the software system and are not directly related to its functionality. The example of such authorities can be the maximum hour of system power-up to power up the customer, the minimum hour of uninterrupted operation of the system, the availability of the graphical interface of the customer, and so on.

Functional and non-functional features of the software system, which is divided into any course project with a variety of recommended topics, are listed below. In the case of lined cases, these actions may be subject to changes in accordance with the course project.

*In any case, it is our responsibility to relate to the specific subject area and specific program being developed.*

#### Functional benefits

1. Creation of a collection of data and recording of new data in a singing format.
2. Reading all the data and the collection of data and their images.
3. Adding a new element of data to the crowd of data.
4. Any element of these data has been updated.
5. You can see any element of this monstrosity.
6. Removal and display of pouch information.
7. Checking the admissibility of the basic data to be entered by the clerk.
8. Providing the customer with the latest information.

#### Non-functional benefits

*Minimal benefits to the customer interface:*

1. The interface of the user may consist of text notifications in Ukrainian (English) that indicate the entry and display of basic and additional data. Vikoristannaya transliteration is blocked.
2. Availability of a text menu for selecting options for the customer's actions.

3. The data that is saved in the data warehouse can be displayed in a tabular view.

When designing the user interface, follow these rules:

Before designing the interface, it is necessary to clearly identify what tasks users will need to use for a particular software product and what capabilities of the interface may fail for them;

The interface may be intuitive and sensible;

The interface can help (and not overwhelm) the user to identify the main components for which the software product is built, and itself:

(1) give clear, understandable instructions; (2) ensure verification and prevention of “incorrect” actions of the clerk; (3) provide reasonable, concise information in situations where it is absolutely necessary;

Follow the standard rules regarding the arrangement of menu items accepted in the operating system, and group menu items in a logical order.

*Vimogs to architecture programs:*

1. There is no less than one data structure from the standard collection library.

2. Viktoristannya flows of data entry and display.

3. Vykorostannaya mechanism vynyatki for processing of pardons.

*Please check the exit code:*

1. Maintaining the principle of encapsulation is equal to access to fields and class methods.

2. The output code of each class of program must be located in the adjacent file.

3. The names of classes, interfaces, class fields, authorities, methods, local variables and constants are also obviously intended to be important in the same subject area.

If the student has chosen the topic of the course project on his own, then you can discuss the program with the teacher and can discuss the results.

In section 1.3 “Mathematical description of the problem”, a mathematical formalization of the problem has been introduced , so that there are clear relationships between quantities that reflect the storage of indicators that need to be resolved based on the input data . In this case, the specifics of the problem at hand may vary across different branches of mathematics and other disciplines. In this way, a mathematical model of the phenomenon is formed with meticulous precision, assumptions and boundaries.

A mathematical model must satisfy two benefits: realism and feasibility.

Under realism, we understand the correct representation of the model of the highest quality of the traced item.

Realization is achieved through reasonable abstraction, through the addition of other details in order to reduce the problem to a problem with known solutions. The rationale is that it is possible to practically spend the necessary payments within an hour, given the available expenditures of the necessary resources.

Therefore, from whom it is necessary to establish mathematical formulas or logical relationships that express the importance of indicators that need to be resolved, from the input data and the necessary explanations.

Another section “Program documentation” consists of four subsections: 2.1 – 2.4. This section contains a description of the architecture of the fragmented software system, information about the testing of the software system and the development of the software product, as well as the quality of the customer service.

Subject 2.1 “Software system architecture.”

A software system means a software program that is being dismantled. This can be a collection of many software components, one or a part of the system.

To reveal the static structure of the software system model, a UML class diagram is used. You can identify, in detail, various relationships between adjacent entities of the subject area, such as objects of the subsystem, and also describe its internal structure and type of communication .

In this section of the explanatory note you must provide:

1. UML diagram of classes that implement the basic business logic of the software system, and description. An example of a description of UML class diagrams was provided by appendix Z.

2. Submissions for listing of programs with output code that represent classes that implement the basic business logic of the software system. The listing of the programs is included in one of the addendums to the explanatory note.

Section 2.2 “Testing the software system”

Testing of a software system is the process of testing the software code and identifying any existing defects. A defect is understood to be a piece of software code, which, when done in the best possible way, leads to an

uncontrollable operation of the system (an action that does not correspond to the forces).

Preliminary testing is the identification of minds in which defects of the system appear, and the recording of these minds.

The metastasis of the software code testing procedure is to minimize the number of defects in the end product.

See the test

### *Modular testing*

During modular testing, the skin module is tested both for compliance with the capabilities and for the presence of problematic sections of the program code that can cause a potential failure in the system.

### *Integration testing*

It is rare for modules to function on their own, so the task that comes after testing the modules is to test the correctness of the interaction between several modules combined as a whole. Such testing is called integration.

### *System testing*

Upon completion of integration testing, all system modules are identical in interfaces and functionality. From this point on, you can move on to system testing, testing the system as a whole as a single object.

The input information for carrying out system testing consists of two classes: functional and non-functional.

Systemic testing is carried out in several stages, in which one type of systemic testing is performed on the skin.

An important type of system testing is functional testing. This type of system testing is intended to confirm that the entire system as a whole is carried out consistently until the completion of the system formalized in the system. During functional testing, all functions of the system are verified from the perspective of human users (both people and other software systems). It is also necessary to verify the functionality of the computer interface and the correctness of the information displayed.

### Documentation of testing procedures

The main purpose of the documentation created during testing is to ensure that the testing process follows all aspects of the testing system with due care.

List of required documentation:

1. Test vimogs.

2. Test plan.
3. This is about testing.

The test cases are divided into system and functional test cases until the end. They clearly describe which aspects of the system are subject to protest in order to ensure its correct functioning, and on the basis of any external effect that can be tested, so that the functionality is verified, implemented correctly.

The test tools will be sufficient to prompt the test plan to verify the software system without knowledge of its program code.

The structure of the test system must maintain the structure of the functional components of the system. As a rule, one system or functional feature is supported by at least one test tool.

For skin testing, it is necessary to avoid the possibility of re-verification - this is possible in the implementation of the system.

On the test platform, a test plan is created - a document that contains a detailed description of how the test will be tested. The test plan describes specific ways to test the functionality of the system.

As a rule, a test plan consists of several test applications, each of which verifies the operation or selection of system functions. For a skin test application, a criterion for successful completion is clearly defined, which can be used to judge the system's compliance with the given requirement.

The structure of the test plan is consistent with the structure of the test case.

The skin point of the test plan is guilty of revenge:

1. The request is made to the effect(s) that is verified by this paragraph;
2. Specifically the meaning of the input data.
3. I am aware of the reaction of the program (text information, meaning of results).
4. Description of the sequence of actions necessary to select points in the test plan.

Following the results of the tests, a report about the test is created. This is the main tool for determining the level of availability of the tested system. Such a statement, at a minimum, is liable to misrepresent information about the skin test test butt and the result of its test (success or failure).

Some test plans are obtained from the results of the testing, providing further information about the reaction of the system and the difference between the results and the results. For example, the description of the skin test pad

provides information about those who have passed the test pad in general. For example, every test plan received as a result will have a column “Test applications passed”, which records the number of passed test applications and the number of test applications passed.

An example of the test plan received from the report about the testing was sent to the assistant I.

*This paragraph of the explanatory note contains a description of the functional testing procedures and their results that are relevant to the specific subject area.*

To describe the testing procedure, we have compiled the following documents:

1. Test vimogs.
2. Test plans received from test reports.

The reports about the testing indicate both positive and negative results of various tests.

However, the final result of the congestion test must be positive, since in the protracted episode it does not indicate any symptoms. For this reason , it is necessary to carry out additional steps to correct the problems and test them. The stench is also to blame for the description in this paragraph of the explanatory note.

Restoring is the process of expanding a ready-made component or component for installation on other computers. In section 2.3 “Introduction to a software product”, it is necessary to provide a description of the hardware and software features necessary for the functioning of the developed software product, and prior to its installation on the customer’s computer.

An example description of the procedure for opening a software product created on the Java SE platform was provided by appendix K.

Subject 2.4 “Koristuvach’s Care” contains the descriptions of the programs, instructions for following the descriptions, and a description to inform the customer that the process of robotic programs may appear.

Clause 2.4.1 “Assignment of the software product” indicates information about its assignment and information sufficient to understand the functions of the software product.

Clause 2.4.2 “Initiation of a software product ” indicates the sequence of actions of the user to ensure the launch, launch and completion of the program, a description of the function, format and possible options for action, in addition

to which types of action are provided. Vikonanny programs, as well as the program's reaction to tsi dii.

In paragraph 2.4.3 "Contact the customer", you can see on-screen notification forms that may appear during the installation of programs, and a description of their replacement.

The butt of the koristuvach's kerivnitstva was brought to the aid of L.

Make an assessment of the positive results of the work (negative ones as well); Possible problems with the results of the work; economic, scientific, social significance of work.

The list of popular sources is a wealth of information that has been quoted, learned or reviewed in the work. Dzherel can be placed in the list in one of the following ways: in the order the message appears in the text, in alphabetical order of the names of the first authors or headings. Vimogi before registration of the list of vikorystnykh dzherel is directed at [1].

The appendices contain the material that is necessary for the completion of the explanatory note, but it cannot be consistently placed in the main part due to great difficulty or methods of creation for other reasons.

Illustrations (business process diagrams, algorithm diagrams, technological processes, dialogue scripts, etc.), tables, intermediate mathematical proofs, formulas and breakdowns, related text and other materials may be issued in the form of additional supplements.

## **5. Follow up before issuing an explanatory note for the course project**

The most important thing when working on a course project is its design before the songs are presented.

When formatting the text of the explanatory note, the traces are in accordance with the state standard of Ukraine DSTU 3008-2015 "Documentation. Issues in the field of science and technology. Structure and design rules" [2]. Further information has been provided before completing an explanatory note for the course project.

The materials of the explanatory note are printed on A4 format paper bottles. Their text is subject to the rules of grammar and style.

The text of the work must be formatted, leaving fields of the following dimensions on the arches: left – 30 mm, right – 10 mm, top – 20 mm, bottom – 20 mm.

The text of the document is to blame buti wiconany s wikiristannyam font Times New Roman (size 14), with a row spacing of 1.2 (37 rows on the side). The smallest font size can be size 10 (this can be selected when submitting the output text to the program). The font must be clear, the text must be black in color and medium in weight. The color scheme is allowed to be used only for drawings (on-screen forms, diagrams, etc.). The strength of the text is due to the fact that it is not true. Verification of the main text is carried out “behind the width” of the page.

The entire text of the explanatory note, including the names of structural elements, is printed in a bold font. Italics and underlining are not allowed.

The paragraph entry is due to the continuation of the entire text and the addition of 1.25 cm.

Mistakes, typos and graphical inaccuracies that appeared during the finalization of the document can be corrected by either being cleaned or coated with white paper and applied to the same place as the corrected text. No more than two corrections are allowed on one page.

When the words and words are shortened, it is necessary to put the first name back, and then put the word in the bows - the shortened word.

The cutaneous structural element of the explanatory note ( crime of subsections, points, subpoints) must begin on a new side.

The names of the elements “ABSTRACT”, “ABSTRACT”, “ENTRANCE”, “CHAPTERS”, “LIST OF QUALIFICATIONS”, “ADDITIONS” should be placed symmetrically to the text (in the center), without paragraph indentation, not numbered (it is not possible to edit “1. INTRODUCTION” " or "ROZDIL 6. VISNOVKI"), enclosed in capital letters without a speck and not underlined.

Headings of subsections, paragraphs and subparagraphs begin in paragraph form and are followed by small letters, except for the first capital, without underlining, without a dash at the end.

Breaks in the middle of words in headings are not allowed.

It is not allowed to place headings and subheadings in the lower part of the page, since there is only one row of text on it.

There are two rows between the title and the back and front text. Stand between the main rows of the heading, and also between the two headings, treat it like the text.

Sections, subsections, paragraphs, subparagraphs must be numbered in Arabic numerals. The sections are marked with serial numbering, for example: 1, 2, 3. They are also responsible for the serial numbering between the sections. The subsection number includes the section number and the serial number of the subsection, which are separated by a dot, for example: 1.1, 1.2, 1.3. The number of the paragraph includes the number of the section, subsection, the serial number of the paragraph, which are separated by a mark, for example: 1.1.1, 1.1.2, 1.1.3.

The sides of the explanatory note of guilt are numbered in Arabic numerals in the upper right corner without a speck. The numbering of pages is from the title column to the remaining text column, including illustrations, tables, and graphs. On the title arch assigned to a course project, the side numbering is not indicated.

The materials contained in the text must be carefully supplemented and supported by illustrations (schemes, drawings, graphs, diagrams). The illustrations are intended to highlight the theme of the course project. Students need to think about what material to illustrate. These are diagrams of classes, diagrams of algorithms, diagrams of information connections , etc.

All illustrations are called drawings, and they are assigned a serial number between the section numbers. The caption of the illustration consists of the word "Figure", the number of the illustration and the name ( for example, "Figure 3.1 - Layout diagram"). Drawings must be wrapped on one side and expanded immediately after the first mention in the text, or on the next page. All illustrations bear references to the text.

The details on the illustration are indicated by the serial number of the illustration, for example, "Fig. 1.2".

In repeated illustrations, the word "marvel " *must be abbreviated*, for example : "(div. Fig. 1.2)."

In this place, where the theme is laid out, connected with the illustration, place the message in the form of a round bow "(Fig. 3.1)" or a fold like: "... as shown in Fig. 3.1".

Write down the skin formula after the first guess in the text from a new row, symmetrically to the text. Skip one row between the formula and text. Formulas are numbered within sections.

Explanations in clear literal terms of the meanings in the formula are provided immediately below the formula. For this purpose, after the formula, put someone and write down the explanation for the skin symbol from the new

row in the same sequence as indicated in the formula, separating the dot from the coma. The first row must begin with a paragraph from the word “de”.

For example,

$$I = UR, \quad (2.1)$$

de U - electrical voltage;

R is the power of the electric current.

If the formula occupies a number of rows, then it can be divided into mathematical symbols: addition, subtraction, multiplication, division and others, which are repeated on the beginning of the next row.

The table must be placed directly after the text, which you can guess first or on the next side. All tables will be responsible. Tables are numbered sequentially within sections. Above the top left corner of the table there is the inscription “That table” with a box containing the serial number and title (for example, Table 1.1 - Structural elements of an explanatory note).

Reflections, if necessary, can be placed in the middle of points or sub-points. Before replanting, install a double fence.

Before the skin position of the transfusion, place a small letter of the Ukrainian abetka with an arch, or, without numbering, a hyphen (the first level of detail).

To further detail the flow of the trace, use Arabic numerals with a bow (another level of detail).

The flows of the first level of detail are divided into small letters from the paragraph indentation, while the other level is in front of the place where the flows of the first level are expanded.

The text of the explanatory note of guilt was sent to literature. When this is done, the serial number of the entries in the square arms is indicated (for example, “... in robots [1-7]”).

Addendums must be completed as a continuation of the explanatory note on the following pages. The skin supplement must start from a new side. The addition is due to the mother's title, which is written in small letters from the first large one, symmetrically in line with the text of the page. In the middle of the row above the title, in small letters with the first great responsibility, the word “Addition” and a large letter are underlined, which means additional (for example, “Addition A”).

Additions are designated successively by the great writers of the Ukrainian alphabet behind the letters Г', Є, І, Ї, И, О, Ч, Ъ. One supplement is designated as supplement A.

## Recommended literature

### Main

1. Bibliographic record. Bibliographic description. Zagalni vimogi and folding rules DSTU GOST 7.1:2006. - K.: Derzhstandart of Ukraine, 2007. - 57 p. [Electronic resource]. - Access mode: [http://lib.khnu.km.ua/storinka\\_vykladacha/oforml\\_spysku/7.1-2006%20%28Ukr%29.pdf](http://lib.khnu.km.ua/storinka_vykladacha/oforml_spysku/7.1-2006%20%28Ukr%29.pdf)
2. Information and documentation. Knowledge from the field of science and technology. Design rules: DSTU 3008-2015. - K.: DP "UkrNDNTs", 2016. - 31 p. [Electronic resource]. - Access mode: [https://science.kname.edu.ua/images/dok/derzhstandart\\_3008\\_2015.pdf](https://science.kname.edu.ua/images/dok/derzhstandart_3008_2015.pdf)
3. Booch G. Object-Oriented Analysis and Design with Applications / G.Booch , RAMaksimchuk et . al . – Boston : Addison-Wesly , 2019. – 717 p.
4. Troelsen A. Pro C# 9 with .NET 5 : Foundational Principles and Practices in Programming / A. Troelsen , P. Japikse – Berkly : Apress , 2021. – 1353 p.

### Dodatkova

5. Lewis J. Java Foundations . Introduction to Program Design and Data Structures / J. Lewis , P. DePasquale , J. Chase – Lancing : Pearson , 2020. – 1104 p.

### Internet resources

1. What are UML diagrams , and how can you use them . [ Electronic resource ]. – Access mode : <https://miro.com/blog/uml-diagram/>
2. C# documentation [ Electronic resource ]. – Access mode : <https://docs.microsoft.com/en-us/dotnet/csharp/>
3. C# Tutorial [ Electronic resource ]. – Access mode : <https://www.tutorialspoint.com/csharp/index.htm>
4. Java Tutorial [ Electronic resource ]. – Access mode : <https://www.w3schools.com/java/default.asp>
5. JDK 20 Documentation [ Electronic resource ]. – Access mode : <https://docs.oracle.com/en/java/javase/20/>

6. Shcherbakov, O. V. Fundamentals of object-oriented programming: beginning . pos\_b . / O. V. Shcherbakov, Yu. E. Parfyonov , V. M. Fedorchenko - Kharkiv: KhNEU im. S. Kuznetsya , 2019. - 236 p. [Electronic resource]. - Access mode: <http://repository.hneu.edu.ua/handle/123456789/23847>

7. Object-oriented programming. Methodical recommendations for laboratory work for students of specialty 121 "Software Security Engineering" of the first (bachelor's) level / setting. Yu. E. Parfionov , O. V. Shcherbakov - Kharkiv: KhNEU im. S. Kuznetsya , 2021. - 92 p. [Electronic resource]. - Access mode: <http://repository.hneu.edu.ua/handle/123456789/27252>

# Add-ons

## Addendum A

A selection of recommended topics for course projects

1. Development of a software product for processing data on the arrival of goods at the enterprise warehouse.
2. Development of a software product for robots with data about physical characteristics.
3. Development of a software product for robots with data about the flow of passenger trains at the Kharkiv station.
4. Development of a software product for robots with data on the triviality of PBX subscribers.
5. Breakdown of a software product for robots with data about the products of a hardware store.
6. Development of a software product for working with data on banking operations.
7. Development of a software product for working with the results of monitoring the fuel consumption at a gas station.
8. Breakdown of the software product for robots with data about postal operations.
9. Breakdown of the software product for robots with data on requests for repair of mobile phones at the service center.
10. Unbundling of a software product for working with contact data.
11. Development of a software product for working with data about postal indexation of the Kharkiv metropolitan area and populated areas in the Kharkiv region.
12. Disaggregation of the software product - the supervisor of the student group.
13. Development of a software product for robots with data on fuel sales at a gas station.
14. Decomposition of a software product for work with data on the hourly use of computers in enterprises.

Continuation of supplement A

15. Development of a software product for robots with information about

the provision of services to subscribers of an Internet provider.

16. Breakdown of the software product for the robot with data about the progress of the robot on a given date.

17. Development of a software product for robots with data about Russian transport services.

18. Development of a software product for robots according to the staffing schedule of the enterprise.

19. Development of a software product for robots with data on passenger transportation.

20. Development of a software product for work with data on book sales in bookstores.

21. Development of a software product for work with data on salary calculations for employees of the enterprise.

22. Development of a software product for robots with data about vitrat paliv at local motor depots.

23. Development of a software product for robots with data on the use of machine hours in the computing center.

24. Development of a software product for robots with data on energy consumption in local factories.

25. Development of a software product for robots with data about the flow of materials in a company's warehouse.

26. Development of a software product for working with data on the income of a business during the current period.

27. The development of a software product for robots with data on the field of work will keep students busy.

28. Breakdown of the software product for robots with data on product deliveries.

29. Development of a software product for robots with data on the transportation of air passengers.

30. Development of a software product for work with data about the hour of work of the enterprise's desktops.

31. Development of a software product for robots with data on the production of parts by robot workers in the workshop.

Completed supplement A

32. Development of a software product for work with data about the

growth of fixed assets of an enterprise.

33. Development of a software product for working with data on the availability of spare parts in a company's warehouse.

34. Development of a software product for work with data on student success.

35. Development of a software product for working with data about the type of payment for telephone services.

36. Development of a software product for working with data about the appearance of cars in a car dealership.

37. Breakdown of the software product for robots with data on the development of Internet resources by clients.

38. Development of a software product for robots with data about automobile spare parts in the warehouse of a car shop.

39. Development of a software product for work with data on the acquisition of literature by students in the library department.

40. Breakdown of the software product for robots with data about the flow of materials in the warehouse.

41. Breakdown of the software product for work with data about applicants who submitted documents for entry before the final deposit.

42. Breakdown of a software product for robots with data on prices for basic food products in the locality.

43. Breakdown of the software product for work with data about the negotiated salaries for military specialists.

44. Breakdown of the software product for robots with data on the production of products by the enterprise.

45. Development of a software product for working with data about industrial enterprises.

46. Development of a software product for robots with data about customer orders.

47. Development of a software product for robots with data about student publications.

Addendum B

Education for the course project

KHARKIV NATIONAL ECONOMIC UNIVERSITY  
NAMED AFTER SEMYON KUZNETS

Faculty of Information Technologies  
Department of Information Systems

ZAVDANNYA

for the course project: Object-oriented programming

2nd year student group 6.04.121.010.21.01

Varenik Grigory Vasilyovich

1. Project topic: “Unbundling of a software system for robot automation with data about <Name of subject area>”
2. The term for the student’s completion of the project “\_\_” \_\_\_\_\_ 202\_ rub.
3. Input data to the project: literary sources, technical documentation for the division of programs, DSTU for the preparation of documentation
4. Place explanatory note:  
Entry Specification for the project. Program documentation. Visnovki. Add-ons.
5. Overflow of graphic material: UML diagram of software system classes.

Calendar plan  
for the course project

Sa lar y no.	The name of the stage is roboti	Rows of viconics behind the plan	Note about Vikonanny
1.	Z'yasuvannya zagalnoj staging zavdannya. The disintegration of the chernetka of the first section of the explanatory note.		
2.	Detailing of the course design task. Clarification of the architecture of the software system. Breakdown of the remaining version of the first section of the explanatory note.		
3.	Software implementation, adjustment and testing of the design.		
4.	Residual lagodzhennya that testuvannya zastosunku. Breaking down the outline of another section of the explanatory note, notes, list of shortlisted articles, appendices, introduction, abstract).		
5.	Separation of the remaining version of the explanatory note and preparation of the electronic presentation		

Date of publication " \_\_ " \_\_\_\_\_ 202\_ rub.

Kerivnyk Ph.D. , Associate Professor Department of IS \_\_\_\_\_  
O.I. Petrenko  
(signature)

The manager was accepted before the viconan \_\_\_\_\_ G.V.  
Dumpling  
(signature)

Illustration of the design of the title arkush

KHARKIV NATIONAL ECONOMIC UNIVERSITY  
NAMED AFTER SEMYON KUZNETS  
DEPARTMENT OF INFORMATION SYSTEMS

## **COURSE Project: OBJECT-ORIENTED PROGRAMMING**

on the topic: “Unbundling of a software system for robot automation with data about <Name of subject area>”

student( s ) 6.04.121.010.21.01 group  
Specialties 121 "Software Engineering"  
security"

first (bachelor's) level

Varenik G.V.

Kerivnyk: Associate Professor of the Department  
of IS, Ph.D. , Associate Professor,  
Petrenko O.I.

National scale \_\_\_\_\_

Number of points: \_\_\_\_\_ Rating: ECTS \_\_\_\_\_

Members of the commission

\_\_\_\_\_  
(signature) (nickname and initials)

\_\_\_\_\_  
(signature) (nickname and initials)

m. Kharkiv – 202\_ rik

Abstract design outline

ABSTRACT

Explanatory note for the course project: 32 pp., 20 figures, 3 tables, 6 pages.

The design objects include functional elements, architecture and code of a software system for working with data about the provision of services to subscribers of an Internet provider.

The method of work is the decomposition of a software system for working with data about the provision of services to subscribers of an Internet provider.

Design method – software products from [app.diagrams.net](http://app.diagrams.net) and Microsoft Visual Studio

The created software system makes it possible to increase the reliability of the service, the reliability and efficiency of decision-making by the Internet provider.

The software product can be used in enterprises to provide access to the Internet.

SUBSCRIBER, LANDSCAPE TRAFFIC, DATA RATE, UML CLASS DIAGRAM, DATA HOLD, .NET, C#

## Instructions for completing an explanatory note

## ZMIST

Entry.....	6
1 Project specification.....	8
1.1 Setting up the task.....	8
1.2 Vimogi before software security.....	10
1.3 Mathematical description of the problem.....	12
2 Software documentation.....	14
2.1 Architecture of the software system.....	14
2.2 Testing the software system.....	16
2.3 Rozlartya of the software product.....	21
2.4 Kerivnitstvo koristuvach.....	23
2.4.1 Assignment to the software product.....	23
2.4.2 Vikorista of the software product .....	24
2.4.3 Congratulations to Koristuvachev.....	28
Visnovki.....	.30
List of vikorista dzherel.....	31
Addendums.....	.....32

## Example of a description of the production of the plant

Acceptance of non-automated processing of apples with vicoristics in a “paper” look. The number of dashboard entries is not limited.

The view has a tabular structure with several components, each of which is called, which corresponds to one of the parameters of any apple: variety (S), color (C), diameter (D), volume (V). The volume of any apple is calculated using formula 1.1:

$$V_i = (1/6) \cdot \pi \cdot D^3, \quad (1.1)$$

where  $i$  is the number of the record entry.

Butt of the filled-in document - div. table 1.1.

Table 1.1

<b>Variety</b>	<b>Kolir</b>	<b>Diameter</b>	<b>Volume</b>
S	C	D	V
Antonivka	Zhovtiy	6	113.04
Jonathan	red gold	10	523.33
...	...	...	...
<b>Average volume of apple</b>			120.99

The journal entry is designed to save the data of an apple, for example:

Antonivka	Zhovtiy	6	113.04
-----------	---------	---	--------

The entrepreneur can add data about the new apple to the list of data by entering the data sheet of the cob data and the data that was covered by formula 1.1 to the relevant fields.

At the end of the reporting period, the industrial scientist calculates the average volume of an apple  $V_{avg}$  using formula 1.2 and enters it in the column "Average volume of an apple" in the spreadsheet:

$$V_{av} = (\sum V_i) / n, \quad (1.2)$$

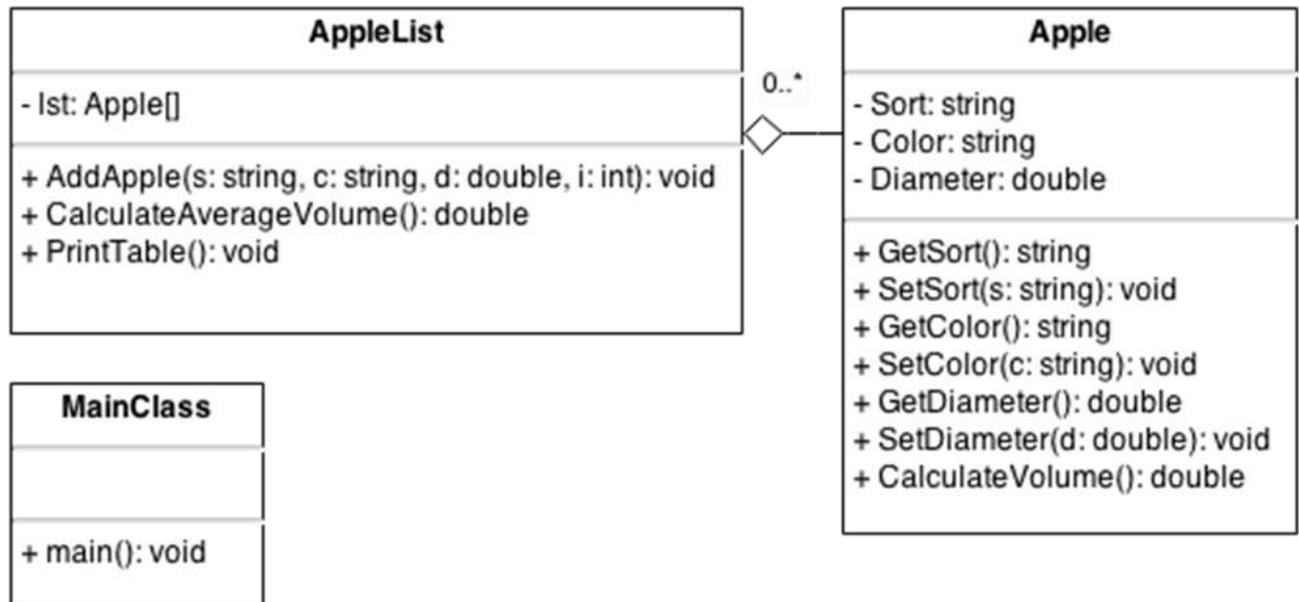
de  $V_i$  – diameter of the i-th apple in the list;

n – number of apples on display.

Develop an object-oriented software system designed to automate the processing of apples in the enterprise.

I will describe the example with UML class diagrams

The program class diagram is shown in Fig. 2.1.



Rice. 2.1 - UML class diagram

The program consists of three classes: Apple , AppleList , MainClass .

Apple class is intended to describe any apple. The Sort field is the apple variety, the Color field is the color of the apple, the Diameter field is the diameter of the apple. The GetSort () method rotates the exact value of the apple variety, the GetDiameter () method rotates the exact value of the apple diameter, the GetColor () method rotates the exact value of the color. The SetSort () method sets a new value for the apple variety, the SetDiameter () and SetColor () methods - the same diameter and color. Method CalculateVolume () - calculates the volume of an apple.

AppleList class is intended to describe the type of apples. There is a basic aggregation between the AppleList and Apple classes: the Ist field of the AppleList class is an array of objects of the Apple type and is intended for storing “apple” objects (so one object can be added to the AppleList class for Apple class objects ).

AddApple () method is required to add an apple object to the lst array .  
Method CalculateAverageVolume () - calculates the average volume of an apple on display. Assigned to the PrintTable () method - “other” value of the fields of the “apple” objects, the diameter of the apple and the bag data of the display.

Class MainClass - head class. The main () method is the entry point to the program. This method is intended for introducing fat into apples display, creating an object for the AppleList class , entering parameters for apples, creating based on the entered parameters for an object for the Apple class , for each apple, adding these objects to the AddApple () method for an object in the AppleL class ist, video tribute to the help of the PrintTable () method of the AppleList class .

Example of the test plan received from the report about  
the testing

**Test stock: No. 1**

Purpose: checking that the software system allows reading data about the services of an Internet provider from a file and correctly displaying them in the customer interface.

**Test cases that are being verified: functional test number 2**

**Change your mind for the test:** the software system must be running, and there is a file with data in the specified format on the computer disk.

**Criterion for passing the test:** the actual behavior of the software system is comparable to that described.

Table H.1

No. salary	Krok script	Result	Rejection result	Note about passing the script (So/No)
1	2	3	4	5
1.	In the program menu, select the "Disable" item	You may receive a notification about entering the path to the file that needs to be opened	A notification appears about entering the path to the file that needs to be opened	So
2.	In the command line, enter the path to the file in the designated format and press the "Enter" button on the keyboard	The console window of the program will display the required data correctly.	The data that was collected from the file is correctly displayed in the console window of the program	So
3.	In the command line, enter the path to the file in an unacceptable format and press the "Enter" button on the keyboard	There may be notifications about those whose data cannot be accessed	There are notifications about those that the data cannot be accessed	So

Completed supplement I

**Note about passing the test (passing/failure):** passing

**Test stock:** No. 2

.....

**Test stock:** No. 3

.....

**Viconano test stocks:** 3

**Test applications completed:** 1

Description of the procedure for opening a software product created  
on the Java SE platform

Access to hardware features:

1. Processor – dual core, 2 GHz or faster
2. Available RAM space - no less than 4 GB
3. Large disk space - no less than 25 GB

Access to software features:

1. Operating system:
  - Windows 10, 11
  - Most current Linux distributions
  - Mac OS X 10.14 or more new
2. Java Runtime Environment 17 or more new

Rozlartiyanya software product on koristuvach's computer  
you can see the autonomous lock:

1. Create a directory on the target disk for storage, for example, MyApp .
2. In the MyApp directory , create a new directory, for example, dist .
3. Copy the newly created jar file (for example, install.jar) to the dist directory .

Java software platform on the Oracle website Runtime Environment of  
the required version and unzip the current archive into a directory, for example  
the appjre folder .

5. Move the appjre directory to the MyApp directory .
6. In the MyApp directory , create a command file for the target operating  
system, for example, start.bat (Windows operating system).

7. Add the following commands to start.bat (Windows operating system):

```
@echo OFF  
set PATH=.\ appjre \ bin  
java - jar dist \install.jar  
pause > NUL
```

1. To check that the program is launched correctly, double-click  
the left mouse button on the start.bat file (Windows operating system).

Addendum L

Butt of kerivnitstva koristuvach (in skorochenni)

## 2.4 Kerivnitstvo koristuvach

### 2.4.1 Software product assignment

The software product is designed to automate the production of apples in the company. It allows you to control the following parameters of the skin: variety, color, diameter and volume; add data about the new apple, calculate the average volume of the skin apple, save data about the calculated number of apples.

### 2.4.2 Vikorista software product

#### Launching programs

Launching programs in an operating system of the Windows family is done in one of the standard ways:

- a) double-click with the left mouse button on the program shortcut;
- b) by clicking the context menu and selecting the "Disable" item;
- c) press the "Start" button on the panel and then select the "All programs" item and then click the left mouse button on the programs shortcut.

#### Basic elements of the software interface

The program's interface consists of the main menu (Fig. 2.1), which appears after launching the program.

```
-----  
Choose the menu item:  
1. Enter initial data  
2. Print inventory  
3. Quit the program  
-----
```

Rice. 2.1 - Golovne program menu

Continuation of addendum L

This menu contains all the commands for Keruvan and Vikon:

1. " Enter" initial data " is intended for the beginning of data entry.
2. " Print " inventory » - to display the list of apples on the console in a table view.
3. Quit the program » - to complete the program.

## Work with the program

### Introduction of cob tributes

To get started, you need to select the main menu item “ Enter” initial data ” by entering the number 1 and then pressing the <Enter> key on the keyboard. After this, a message is consistently displayed on the screen to enter the number of apples in the pan, as well as the variety, color and diameter of the skinned apple (Fig. 2.2). After entering the value of the skin parameters, you must press <Enter> .

```
1
Enter number of apples: 2
Enter apple sort: АНТОНІВКА
Enter apple color: ЖОВТИЙ
Enter apple diameter: 4.6
Enter apple sort: Джонатан
Enter apple color: Червоний
Enter apple diameter: 5
```

Rice. 2.2 - Data entry interface

### Displaying the list of apples on the console

For this purpose, vibrate the main menu item “ Print” inventory » Enter the number 2 and press the <Enter> key on the keyboard.

After completing the actions, a list of apples will appear on the computer screen in a table form (Fig. 2.3). For each of them, the table displays the values of variety, color , diameter and calculated volume. In the “ Total ” row, the program can be used to increase the average apple volume values shown in the table.

Completed addendum L

Please note that after selecting this item in the main menu, the table appears only if you have used the program in advance to obtain the cob data of at least one apple.

2

Sort	Color	Diameter	Volume	
Антонівка	Жовтий	4.60	50.97	
Джонатан	Червоний	5.00	65.45	

Total: 58.21

Rice. 2.3 - List of apples in a table form

Completed work with program

For this you need to select the menu item "Quit" the program " (number 3) and press <Enter> .

### 2.4.3 Congratulations to Koristuvacheva

Yakshcho koristuvach program by selecting the item "Print" inventory » of the main menu, without entering the initial data, the following notification will appear on the computer screen (Fig. 2.4):

```
2  
No data to print!
```

Rice. 2.4 - Please note if data is available for display

## Contents

Entry.....	3
1. Meta department of course project .....	4
2. Organization of course project.....	5
3. Structure and obsyag course project .....	6
4. Methodical recommendations for the breakdown of the structural elements of the explanatory note for the course project.....	6
5. Before completing the explanatory note for the course project.....	16
Recommended literature.....	19
Addendums..... .....	21

EDUCATIONAL EDITION

# **OBJECT-ORIENTED PROGRAMMING**

**Guidelines  
before completing a course project  
for advanced speciality 121  
"Software Engineering" of the educational program  
"Software engineering" of the first (bachelor's) level**

Authors: Parfionov Yu.E.  
Grizun L.E.

D. O. Bondarenko is responsible for the publication

Editor

Proofreader

Plan 2024 Pos. No. 78 EB.

*Certificate of entry of the subject of the publishing business into the State Register, Dk  
No. 4853 dated February 20, 2015.*