**Yuri Fedkovich Chernivtsi National University**

(full name of higher education institution)

**Educational and Scientific Institute of Physical, Technical and Computer Sciences**

(name of the institute/faculty)

**Chaircomputer systems and networks**

(name of department)

# SYLLABUS

**academic discipline**

# *Design technologies of software systems*

(indicate the name of the academic discipline (foreign, if the discipline is taught in a foreign language))

**Mandatory**

(mandatory or optional)

**Educational and scientific program - "Computer engineering of technologies**

***Internet of Things and Cyber-Physical Systems"***

**Specialty 123 – Computer engineering**

(code and name of specialty)

**Branch of knowledge 12 – Information technologies**

(code and name of field of knowledge)

**Level of higher education – second (master's)**

(specify: first (bachelor's)/second (master's)/third (educational and scientific))

***Educational and Scientific Institute of Physical, Technical and Computer Sciences***

(name of the faculty / institute at which specialists are trained according to the specified educational and professional program)

**The language of instruction is Ukrainian, English**

(the language in which the discipline is read)

Developers:Dvirnychuk Kostyantyn Vasyliovych, assistant of the Department of Computer Science, candidate. physics and mathematics sciences,

(specify the authors (teacher(s)), their positions, scientific degrees, scientific titles)

**Teacher profile(s)** https://csn.chnu.edu.ua/, https://csn.chnu.edu.ua/employees/dvirnychuk-kostyantyn-vasylovych/

**Contact phone** +(38) 0372 50 94 32 (CSM department) - K.V. Dvirnychuk.

**Email:** k.dvirnychuk@chnu.edu.ua

**Course page in Moodle** https://moodle.chnu.edu.ua/course/view.php?id=3145

**Consultations** *online:Monday from 17.00 to 18.00*
*Face-to-face consultations: Thursday from 17.00 to 18.00*

**1. Abstract of the discipline**

The course "Software Systems Design Technologies" is designed to provide skills in creating software applications using software design templates (patterns), to be able to test developed software applications, and to understand the basic principles of software architectural templates.

Studying this selective discipline provides students with a number of advantages, since modern programming cannot be imagined without designing the structure of the program itself based on architectural templates and design patterns. In particular, one of the popular architectural template solutions is the MVC template, the basics of which are presented in the framework of this discipline.

**2. The purpose of the educational discipline:** to provide students with systematic knowledge of the purpose, tasks and principles of software system design technologies. To teach students to use design templates (patterns) in their software development. Explain to students the essence of architectural patterns and understand MVC programming in more detail. To provide an opportunity for students to master some modern technologies and to provide them with the concepts and approaches of testing and debugging programs.

**The task is**on the basis of the acquired theoretical knowledge, develop students' ability to develop their own software products, use design templates in their programs, build MVC-models of programs, test programs. Create software applications using modern technologies.

**3. Prerequisites**For kFor correct understanding and mastering of the material of this course, students must first complete courses in: Programming, Object-oriented programming, Software engineering.

**4. Learning outcomes**

As a result of studying the academic discipline, the student should

**Know:**programming technologies, design patterns, structure of MVC-models of programs, methods of testing software developments.

**Be able:**build software applications, use design templates and architectural templates for this, test software applications.

**Acquire competences**:

*ZK - general*

3К1. Ability to adapt and act in a new situation.
3К2. Ability to abstract thinking, analysis and synthesis.
3К3. Ability to conduct research at an appropriate level.
3К4. Ability to search, process and analyze information from various sources.
3К5. Ability to generate new ideas (creativity).
3К6. Ability to identify, pose and solve problems.
3К7. Ability to make informed decisions.

*SK - professional (special)*

СК1. Ability to determine technical characteristics, design features, application and operation of software, software and technical tools, computer systems and networks of various purposes.

СК2. Ability to develop algorithmic and software, components of computer systems and networks, Internet applications, cyber-physical systems using modern methods and programming languages, as well as design automation tools and systems.
СК3. Ability to design computer systems and networks taking into account objectives, constraints, technical, economic and legal aspects.
СК4. Ability to build and research models of computer systems and networks.
СК5. Ability to build architecture and create system and application software of computer systems and networks.
SK12. The ability to solve computer engineering tasks using hardware and software data processing, artificial intelligence tools, cloud technologies, the Internet of Things, and computerized information and measurement systems and complexes.

*PRN - program learning outcomes*

PH1. Apply general approaches to cognition, methods of mathematics, natural and engineering sciences to solving complex problems of computer engineering.
PH2. Find the necessary data, analyze and evaluate them.
PH3. Build and research models of computer systems and networks, evaluate their adequacy, determine limits of applicability.
PH4. Apply specialized conceptual knowledge, including modern scientific achievements in the field of computer engineering, necessary for professional activity, original thinking and conducting research, critical thinking of information technology problems and at the border of the fields of knowledge.
PH5. Develop and implement projects in the field of computer engineering and related interdisciplinary projects taking into account engineering, social, economic, legal and other aspects.
PH6. Analyze problems, identify and formulate specific problems that need to be solved, choose effective methods of solving them.
PH8. Apply knowledge of technical characteristics, design features, purpose and rules of operation of software and technical means of computer systems and networks to solve complex problems of computer engineering and related problems.
PH9. Develop software for embedded and distributed applications, mobile and hybrid systems.
PH10. Search for information in various sources to solve computer engineering problems, analyze and evaluate this information.
PH11. Make effective decisions regarding the development, implementation and operation of computer systems and networks, analyze alternatives, assess risks and likely consequences of decisions.

## 5. Description of the academic discipline
### 5.1. general information

| **The name of the academic discipline** OK09 Design technologies of software systems | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Form of education** | **A year of training** | **Semester** | **Number** | | | **Number of hours** | | | | | | **Kind in conclusion control** |
| | | | **loans** | **hours** | **content modules** | **lectures** | **practical** | **seminar** | **laboratory** | **independent work** | **individual tasks** | |
| **Full-time** | **1** | **1** | **3** | **90** | **1** | **15** | **-** | **-** | **15** | **60** | **-** | **exam** |

**Note**. The ratio of the number of hours of classroom classes to independent and individual classes work is: for full-time education - 0.5 (30/60);

### 5.2. Didactic map of the academic discipline

| Names of content modules and topics | Number of hours | | | | | |
|---|---|---|---|---|---|---|
| | everything | including | | | | |
| | | l | p | lab | ind | s.r. |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **Content module 1**. TPPS | | | | | | |
| Topic 1. The concept of software architecture. | 30 | 3 | | 7 | | 20 |
| Topic 2. Patterns of software design. | 32 | 8 | | 4 | | 20 |
| Topic 3. Testing and debugging of software systems. | 28 | 4 | | 4 | | 20 |
| Together according to content module 1 | 90 | 15 | | 15 | | 60 |
| **Only hours** | 90 | 15 | | 15 | | 60 |

### 5.3. Topics of seminar or practical or laboratory classes

| No s/p | Topic name | Number hours |
|---|---|---|
| 1 | Analysis of the software architecture template | 3 |
| 2 | MVC application development | 4 |
| 3 | Software design templates (patterns). | 4 |
| 4 | Software testing | 4 |
| | Together | **15** |

### 5.4. Subjects of individual tasks

This course does not include individual tasks.*

* INDZ - may be recommended in individual cases for students who have successfully mastered the basic educational material, for the purpose of in-depth study or improvement of the materials of a certain content module, or in general for the academic discipline by the decision of the department or teacher.

### 5.5. Independent work

| No s/p | Topic name | Number hours |
|---|---|---|
| 1 | CMF and CMS | 10 |
| 2 | Development of HMVC, MVVM, NA, MVP, PAC models of programs | 10 |
| 3 | Antipatterns | 10 |
| 4 | Generating patterns | 10 |
| 5 | Classification and testing | 20 |
| | Together | **60** |

## 6. Forms and methods of education

**Forms of education**- these are problem and review lectures, laboratory classes, classes with the use of computer and telecommunication equipment, interactive classes with the teaching of one student by others, integrated classes, problem classes, video lectures, video classes and video conferences using Google Meet, Zoom, Cisco Webex, classes with using the Moodle e-learning system.

**Methods:**problem presentation of the material, partially research and research laboratory practices, presentations, case studies, consultations and discussions, work in the online classroom: electronic lectures, laboratory work, remote consultations, etc., aimed at activating and stimulating the educational and cognitive activities of students .

**Approaches to learning**: student-centered, problem-oriented, activity-oriented, communicative, professional-oriented, interdisciplinary approaches are used.

**Implementation of the educational process**is carried out during lectures, laboratory classes, independent out-of-class work using modern educational information technologies, consultations with teachers.

The following training methods are used to develop skills and abilities:

- verbal/verbal (lecture, explanation, story, conversation, instruction);
- visual (observation, illustration, demonstration);
- practical (conducting an experiment, practices);
- explanatory-illustrative or information-receptive, which involves the presentation of ready-made information by the teacher and its assimilation by students;
- reproductive (performance of laboratory tasks according to the sample);
- method of problem presentation of the material in lectures.

**Hardware and software/hardware.**

Computers in computer classes No. 307, No. 311, No. 312, No. 313 8k. ChNU - Department of KSM.

Software: there are no special software requirements. Laboratory work can be performed on any operating system and in any software development environment.

.

## 7. Control and evaluation system

### 7.1. Distribution of the maximum possible number of points that students receive for performing all types of educational activities

**Content module 1.**

T1. The concept of software architecture (performance of laboratory works #1-2 – 20 points)

T2. Software design templates (performance of laboratory work #3 – 15 points)

T3.Software application testing(performance of laboratory works #4 – 15 points)

M. Modular control work - 10 points

**Rating scale: national and ECTS**

| The sum of points for all types of educational activities | ECTS assessment | Evaluation on a national scale | |
|---|---|---|---|
| | | for an exam, course project (work), practice | for credit |
| 90 - 100 | **AND** | perfectly | counted |
| 80-89 | **IN** | fine | |
| 70-79 | **WITH** | | |
| 60-69 | **D** | satisfactorily | |
| 50-59 | **IS** | | |
| 35-49 | **FX** | unsatisfactory with the possibility of reassembly | not counted with the possibility of reassembly |
| 0 - 34 | **F** | unsatisfactory with mandatory repeated study of the discipline | not counted with mandatory repeated study of the discipline |

## 8. Evaluation tools

The means of evaluating the student's learning outcomes are: tasks for performing laboratory works, as well as modular control works.

### Forms of current and final control

The forms of current control of the level of knowledge are the student's oral and written answer when defending the completed laboratory work, as well as a written answer when writing modular control papers.

Forms of the final control of the level of knowledge are the student's oral and written answer when passing the test.

## 9. Politics of discipline.

It is determined by the system of requirements of the teacher regarding the level of knowledge and assimilation of the material by the student when studying the discipline, and is based on the principles of academic integrity, taking into account the norms of the legislation of Ukraine on academic integrity and the Statute, regulations of the University, and other regulatory documents that regulate the organization of the educational process when studying the discipline.

The requirements relate to incentives and the awarding of additional points for active participation in discussions regarding the analysis and discussion of thematic material in lectures and laboratory classes, thorough preparation for classes, absence of absences without valid reasons, detection of in-depth knowledge during the defense of laboratory workshop reports and modular control.

## 10. Recommended literature

### 10.1. Basic (basic)


1. Dvirnychuk K.V. Synopsis of lectures on the educational discipline "Technologies of designing software systems" (electronic edition). – Chernivtsi, 2022. – 100 p.
2. Budai A.Design patterns are as simple as doors. - electronic resource, 2019. - 90 p.
3. Head First book. Design patterns / Eric Freeman, Elizabeth Robson, Burt Bates, Kathy Sierra. – Fabula, 2020. – 672 p.
4. Borodkina I. Software engineering. Handbook for students of higher educational institutions / I. Borodkina, G. Borodkin. – Center for educational literature, 2018. – 204 p.
5. Yu. Hrytsyuk. Analysis of software requirements / Yu. Hrytsyuk. – Lviv Polytechnic, 2018. – 456 p.
6. Melnyk N. Introduction to software engineering / N. Melnyk, E. Levus – Lviv Polytechnic, 2018. – 248 p.
7. C# programming language [electronic resource]. Access mode:https://abitap.com/category/c/
8. Design patterns [electronic resource]. Access mode: https://abitap.com/category/paterny-proektuvannya/.


### 10.2. Auxiliary


1. Design patterns [electronic resource]. Access mode: https://metanit.com/sharp/patterns/.
2. C# programming language [electronic resource]. Access mode: https://metanit.com/sharp/tutorial/.
3. Troelsen E. The C# 6.0 programming language and the .NET 4.6 platform / E. Troelsen, F. Jepiks. - Book bench, 2019. - 800 p.
4. Robert C. Martin. Pure architecture / Robert S. Martin. – Fabula, 2019. – 368 p.
5. Robert C. Martin. Clean code / Robert C. Martin. – Fabula, 2019. – 416 p.
6. Melnyk R.A. Programming of web applications (front-end and back-end) / R.A. Miller. – Lviv Polytechnic, 2018. – 248 p.


## Information resources

1. https://csn.chnu.edu.ua/about-us/ok-rivni/
2. https://csn.chnu.edu.ua/spetsialnist-123-komp-yuterna-inzheneriya-onp-komp-yuterna-inzheneriya-tehnologij-internetu-rechej-ta-kiberfizychnyh-system-magistratura-2-r/