

**НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ**  
**ІНСТИТУТ ПРОБЛЕМ МАТЕМАТИЧНИХ МАШИН І СИСТЕМ**

**ЗАТВЕРДЖЕНО**  
рішенням вченої ради  
ІПММС НАН України  
від « 03 » серпня 2022 року  
протокол № 7

**РОБОЧА НАВЧАЛЬНА ПРОГРАМА**  
**ДИСЦИПЛІНИ**  
***«ТЕХНОЛОГІЯ СТВОРЕННЯ ПРОГРАМНОГО***  
***ЗАБЕЗПЕЧЕННЯ»***

Третій освітньо-науковий  
рівень вищої освіти – доктор філософії

Спеціальність – 122 Комп'ютерні науки

Київ 2022

## I. ЗАГАЛЬНІ ВІДОМОСТІ

**Предметом** вивчення навчальної дисципліни є правила, способи, методи та засоби розробки програмних продуктів.

**Міждисциплінарні зв'язки.** Дисципліна базується на знаннях з методів та засобів комп'ютерних технологій та використовуватиметься при програмуванні.

**Метою** викладання навчальної дисципліни «Технологія створення програмного забезпечення» є засвоєння аспірантами теоретичних основ сучасних технологій створення програмних продуктів та практичних навичок їх реалізації.

## II. РОЗПОДІЛ УЧБОВОГО ЧАСУ

| Семестр            | Семестрова атестація | Всього | Розподіл за семестрами та видами занять |                |          |             |                   |
|--------------------|----------------------|--------|---|----------------|----------|-------------|-------------------|
|                    |                      |        | Лекції                                  | Практ. заняття | Семінари | Лаб. роботи | Самостійна робота |
| 1                  | Диф. залік           | 60     | 20                                      | -              | -        | -           | 40                |
| Кількість кредитів |                      | 2      |   |                |          |             |                   |

Перелік основних компетенцій, що мають бути набути протягом навчання наведено в таблиці 1.

Таблиця 1

| Обов'язкові компетентності   |  | Результати навчання  |
|--|--|--|
| Загальні компетентності  | Спеціальні (фахові) компетентності   |  |
| <b>ЗК02.</b> Здатність до пошуку, оброблення та аналізу інформації з різних джерел.<br><b>ЗК03.</b> Здатність працювати в міжнародному контексті | <b>СК02.</b> Здатність застосовувати сучасні методології, методи та інструменти експериментальних і теоретичних досліджень у сфері комп'ютерних наук, сучасні цифрові технології, бази даних та інші електронні ресурси у науковій та освітній діяльності. | <b>РН01.</b> Мати передові концептуальні та методологічні знання з комп'ютерних наук і на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з |

|  |   |   |
|--|---|---|
|  | <p><b>СК06.</b> Здатність аналізувати та оцінювати сучасний стан і тенденції розвитку комп'ютерних наук та інформаційних технологій</p> | <p>відповідного напрямку, отримання нових знань та/або здійснення інновацій.</p> <p><b>РН04.</b> Розробляти та досліджувати концептуальні, математичні і комп'ютерні моделі процесів і систем, ефективно використовувати їх для отримання нових знань та/або створення інноваційних продуктів у комп'ютерних науках та дотичних міждисциплінарних напрямках.</p> <p><b>РН07.</b> Розробляти та реалізовувати наукові та/або інноваційні інженерні проекти, які дають можливість переосмислити наявне та створити нове цілісне знання та/або професійну практику і розв'язувати значущі наукові та технологічні проблеми комп'ютерної науки з дотриманням норм академічної етики і врахуванням соціальних, економічних, екологічних та правових аспектів</p> |
|--|---|---|

### III. ЦІЛІ ТА ЗАДАЧІ ДИСЦИПЛІНИ

Основними завданнями вивчення дисципліни “Технологія створення програмного забезпечення” є:

- формування систематизованого уявлення щодо концепцій, моделей, та інструментальних засобів сучасних технологій розробки програмного забезпечення складних систем;
- одержання практичної підготовки в сфері вибору та застосування технологій програмування для задач створення інформаційних систем.

Згідно з вимогами освітньо-професійної програми студенти повинні:

**знати:** основні методи, засоби та прийоми реалізації фаз життєвого циклу розробки програмних продуктів;

**вміти:** реалізовувати основні етапи проектування програмного продукту на основі сучасних технологій, застосовувати сучасні методології, методи та інструменти творення програмного забезпечення у науковій та освітній діяльності.

**мати навички** розробки, тестування, налаштування і документування програм.

## IV. ТЕМАТИЧНИЙ ПЛАН

### IV.1. Розподіл учбового часу по темах

| Назва розділів, тем  | Розподіл за семестрами та видами занять |               |                      |                 |                     |                    |                         |
|--|---|---------------|----------------------|-----------------|---------------------|--------------------|-------------------------|
|  | Всього годин                            | Лекції, годин | Практ. заняття годин | Семінари, годин | Лаб. роботи, годин, | Комп. практ, годин | Самостійна робота годин |
| 1  | 2                                       | 3             | 4                    | 5               | 6                   | 7                  | 8                       |
| Семестр 1  |   |               |                      |                 |                     |                    |                         |
| <b>Модуль 1. Життєвий цикл і стандарти ПЗ</b>  |   |               |                      |                 |                     |                    |                         |
| <u>Тема 1.</u> Поняття програмного забезпечення та проблеми розробки складного ПЗ. Еволюція технологій програмування. Структурне, модульне, об'єктне та компонентне програмування  | 6                                       | 2             |                      |                 |                     |                    | 4                       |
| <u>Тема 2.</u> Поняття життєвого циклу ПЗ. Основні, допоміжні та організаційні процеси життєвого циклу. Взаємозв'язок між процесами. Моделі життєвого циклу ПЗ. Моделі послідовного виконання стадій (каскадна, каскадна із зворотним зв'язком, каскадна з прототипуванням). Ітераційні моделі (інкрементна, спіральна, модель еволюційного прототипування)  | 6                                       | 2             |                      |                 |                     |                    | 4                       |
| <u>Тема 3.</u> Міжнародні та національні стандарти розробки складних програмних продуктів. Нормативно-методичне забезпечення створення ПЗ. Стандарти процесів життєвого циклу (ISO/IEC 12207:1995, ДСТУ 3918 – 1999)   | 6                                       | 2             |                      |                 |                     |                    | 4                       |
| <b>Модуль 2. Методи та засоби розробки ПЗ</b>  |   |               |                      |                 |                     |                    |                         |
| <u>Тема 4.</u> Структурний підхід до проектування ПЗ. Функціональні діаграми і діаграми потоків даних – призначення, особливості, порівняльний аналіз. Моделі і технології SADT, IDEF3, DFD, ERM, IDEF1X. Об'єктний підхід до проектування ПЗ. Основні принципи будування і поняття об'єктної моделі. Уніфікована мова моделювання UML. Діаграми UML – призначення, види, особливості. Технології RUP, MSF, RAD, XP та ін. | 6                                       | 2             |                      |                 |                     |                    | 4                       |

|   |   |   |  |  |  |  |   |
|---|---|---|--|--|--|--|---|
| <u>Тема 5.</u> Архітектура ПЗ, стандарти опису архітектури ПЗ. Поняття архітектури ПЗ в рамках структурного та об'єктного підходів. Структурні карти Константайна, Джексона. Проектування архітектури ПЗ в UML. Патерни проектування ПЗ. Повторне використання проектних рішень і поняття патерну. Склад патерну – проблема, рішення, наслідок. Представлення патерну в мові UML. Переваги використання патернів  | 6 | 2 |  |  |  |  | 4 |
| <u>Тема 6.</u> Засоби автоматизації розробки програмних продуктів. CASE – технології розробки програмних продуктів при структурному і об'єктному підходах. CASE – засоби BPWin, ERWin, CASE Analitic, Designer/2000, SilverRun. CASE – засоби Rational  | 6 | 2 |  |  |  |  | 4 |
| <b>Модуль 3. Вимоги замовника та якість ПЗ</b>  |   |   |  |  |  |  |   |
| <u>Тема 7.</u> Аналіз вимог замовника до ПЗ при структурному (процесному) підході до проектування. Моделі етапу аналізу – функціональні діаграми, діаграми потоків даних, діаграми відносин компонентів. Форми представлення специфікацій-блок-схеми, псевдокоди, природномовні описи. Аналіз вимог до ПЗ при об'єктному підході. Діаграми прецедентів (варіантів використання), діаграми класів, діаграми послідовності дій, діаграми діяльності. Документування специфікацій в мові UML | 6 | 2 |  |  |  |  | 4 |
| <u>Тема 8.</u> Якість процесів створення ПЗ – сертифікація і оцінка. Модель оцінки зрілості процесів CMM – рівні зрілості, методи оцінки SPA, SCE, інтегрована модель ICMM. Стандарти ISO 9000:2000, ISO 9001:2001, ISO 9004:2000. Методика SPMN. Якість програмного забезпечення. Основні характеристики якості – функціональність, надійність, практичність, ефективність та ін. Метрики основних характеристик якості. Інструменти аналізу якості                                      | 6 | 2 |  |  |  |  | 4 |

|  |    |    |  |  |  |  |    |
|--|----|----|--|--|--|--|----|
| <i>Тема 9.</i> Призначення і зміст процесу верифікації. Основні дії та перевірки в процесі верифікації. Призначення і зміст процесу валідації (атестації). Основні дії процесу валідації. Ступені незалежності процесів верифікації та валідації<br>Тестування – призначення і зміст. Стадії тестування. Підходи до формування тестових наборів- структурний підхід (принцип «скляної або білої скринька») і функціональний підхід (принцип «чорної скриньки»). Методи тестування.<br>Стандарти тестування | 3  | 1  |  |  |  |  | 2  |
| <i>Тема 10.</i> Випробування і супровід програмних продуктів. Динаміка відносної вартості процесів супровіду ПЗ. Коректуючий, вдосконалюючий та адаптуючий супровід. Основні дії процесу супровіду. Випробування ПЗ  | 3  | 1  |  |  |  |  | 2  |
| <b>Модуль 4. Документування та маркетинг ПЗ</b>  |    |    |  |  |  |  |    |
| <i>Тема 11.</i> Експлуатаційна, операційна, рекламна документація на ПЗ. Стандарти документування ПЗ. Пояснювальна записка, керівництво користувача, керівництво системного програміста. Основні правила оформлення програмної документації. Маркетинг програмних продуктів. Особливості маркетингу. Захист програмних продуктів від несанкціонованого використання. Приклади маркетингу ПЗ відомих виробників.  | 6  | 2  |  |  |  |  | 4  |
| Всього за 1 семестр  | 60 | 20 |  |  |  |  | 40 |

## V. ПОТОЧНИЙ ТА ПІДСУМКОВИЙ КОНТРОЛЬ

Поточний контроль здійснюється під час проведення лекцій та практичних занять.

Підсумковий контроль – це оцінювання засвоєння студентами всього теоретичного матеріалу та рівня практичної підготовки з навчальної дисципліни.

Підсумкова оцінка виставляється за результатами поточного контролю за шкалою оцінювання, наведеною в таблиці 2.

**Таблиця 2**

| <b>Оцінка (за національною шкалою)</b>     | <b>Бали</b> |
|--|-------------|
| Атестований з оцінкою<br>"відмінно"        | 91-100      |
| Атестований з оцінкою<br>"добре"           | 76-90       |
| Атестований з оцінкою<br>"задовільно"      | 60-75       |
| Не атестований з оцінкою<br>"незадовільно" | 26-59       |
| Не атестований з оцінкою<br>"н/а"          | 0-25        |

Підсумковий контроль проводиться у формі диференційованого заліку.

## **VI. ПЕРЕЛІК ЕКЗАМЕНАЦІЙНИХ ПИТАНЬ**

### **вопросы №1**

1. Кризис 60-х годов в программировании.
2. Основные этапы развития технологий программирования.
3. Принципы этапа структурного программирования.
4. Понятия «программа», «программное средство», «программный продукт», «программное обеспечение».
5. Этапы создания компьютерной программы.
6. Процедурное программирование. Свойства и основные языки.
7. Объектно-ориентированное программирование. Свойства и основные языки.
8. Компонентное программирование.
9. Модульное программирование
10. Характеристики и проблемы крупномасштабных проектов программного обеспечения.
11. Сопровождение программного средства. Относительная трудоемкость.
12. Нормативно-методическое обеспечение создания ПО. Общее назначение и состав.
13. Стандарт жизненного цикла ПО. Назначение и общая структура.
14. Международный стандарт ISO/IEC 12207:1995 Виды процессов.
15. CASE-средства проектирования ПО. Назначение, классификация.
16. Языки программирования семейства С (С, С++, С#). Общая характеристика.
17. Язык программирования JAVA. Общая характеристика.
18. Язык программирования BASIC. Общая характеристика.
19. Среда и язык программирования Delphi. Общая характеристика.
20. Классификация программных продуктов по функциональному признаку.



## вопросы №2

1. «Процесс верификации» (В.4) жизненного цикла ПО. Назначение, действия.
2. «Процесс аттестации (валидации)» (В.5) жизненного цикла ПО. Назначение, действия.
3. «Процесс обеспечения качества» (В.3) жизненного цикла ПО. Назначение, действия, задачи.
4. Модель жизненного цикла ПО. Общие понятия.
5. Каскадная модель жизненного цикла ПО. Сущность и основные этапы.
6. Каскадная модель жизненного цикла ПО. Достоинства и недостатки.
7. Спиральная модель жизненного цикла ПО. Принципиальные особенности.
8. Спиральная модель жизненного цикла ПО. Достоинства и недостатки.
9. Итерационная модель жизненного цикла **RAD**.
10. Характеристика функциональности ПО. Определение и пример подхарактеристик.
11. Характеристика практичности ПО. Определение и пример подхарактеристик.
12. Характеристика надежности ПО. Определение и пример подхарактеристик.
13. Характеристика переносимости (мобильности) ПО. Определение и пример подхарактеристик.
14. Характеристика эффективности ПО. Определение и пример подхарактеристик.
15. Модель CMM. Уровни зрелости процессов.
16. Модель CMM. Назначение, достоинства и недостатки.
17. Методика SPMN (CBP). Назначение, особенности, рекомендуемые «навыки».
18. Сертификация программных средств. Цель, основные подходы.
19. Сертификация программных средств в Украине.
20. Стандарты в области систем менеджмента качества.

## вопросы №3

1. Моделирование бизнес-процессов и ПО. Общее назначение моделей, виды моделей.
2. Базовые принципы методов и моделей при структурном подходе.
3. Виды моделей в структурном анализе и проектировании.
4. SADT-модели. Базовый функциональный блок.
5. SADT-модели. Декомпозиция диаграмм.
6. Модели IDEF3. Базовый функциональный блок.
7. Модели IDEF3. Виды соединений блоков на диаграммах.
8. Модели DFD. Основные компоненты диаграмм.
9. Модели DFD. Последовательность построения иерархии процессов.
10. Модели DFD. Критерии завершения декомпозиции и создания спецификаций процессов.
11. Сравнительный анализ моделей SADT и DFD.
12. Модели ERM. Сущности и атрибуты.
13. Модели ERM. Связи.
14. Базовые принципы построения объектных моделей. Классы и объекты
15. Язык UML. Назначение, группы и виды диаграмм.
16. Язык UML. Диаграмма прецедентов.
17. Язык UML. Диаграмма классов и объектов.
18. Язык UML. Диаграмма последовательностей.
19. Язык UML. Диаграмма деятельности.
20. Сравнение структурного и объектного подходов к моделированию.

#### **вопросы №4**

1. Цель и содержание предпроектных исследований при проектировании ПО.
2. Факторы, определяющие свойства проектируемого ПО.
3. Цель и содержание ТЗ на ПО.
4. Последовательность разработки ТЗ на ПО.
5. Выбор типа архитектуры при эскизном проектировании ПО.
6. Выбор типа пользовательского интерфейса при эскизном проектировании ПО.
7. Выбор языков и среды программирования при эскизном проектировании ПО.
8. Выбор общего подхода (структурного, объектного) к проектированию ПО.
9. Функциональные диаграммы проектируемого ПО (эскизный проект).
10. Диаграммы потоков данных проектируемого ПО (эскизный проект).
11. Структурная схема разрабатываемого ПО (техн. проект).
12. Функциональная схема разрабатываемого ПО (техн. проект).
13. Метод пошаговой детализации разрабатываемого ПО (техн. проект).
14. Структурные карты Константайна. Назначение и составные элементы.
15. Описание структурных алгоритмов с помощью псевдокодов.
16. Описание структурных алгоритмов с помощью Flow-форм
17. Общая методика отладки ПО (лаб 11,12)
18. Документация ПО. Виды программных документов. (лаб 11,12)
19. Структурный подход к тестированию ПО («стеклянный ящик»).
20. Функциональный подход к проектированию ПО («черный ящик»).

#### **вопросы №5**

1. Понятие технологичности ПО; факторы, определяющие технологичность.
2. Эволюция понятия «модуль» ПО, основные общие требования к модулям.
3. Сцепление модулей по данным.
4. Сцепление модулей по образцу.
5. Сцепление модулей по управлению.
6. Сцепление модулей по общей области данных.
7. Сцепление модулей по содержимому.
8. Функциональная связность модулей .
9. Последовательная связность модулей .
10. Информационная связность модулей .
11. Процедурная связность модулей .
12. Временная связность модулей .
13. Логическая связность модулей .
14. Нисходящий подход в разработке ПО. Достоинства и недостатки.
15. Восходящий подход в разработке ПО. Достоинства и недостатки.
16. Базовые алгоритмические структуры структурного программирования.
17. Факторы, определяющие ясность и читабельность программ.
18. Факторы, определяющие вычислительную эффективность программ.
19. Стиль оформления текстов модулей.
20. Общие правила оформления программных модулей.

## VII . НАВЧАЛЬНО-МЕТОДИЧНА ЛІТЕРАТУРА

### Основна література

1. Інформаційно-обчислювальні комплекси та АСУ.: Навчально-методичний посібник./уклад Хлобистова О.А., Гладка М.В., Бобрівник К.Є – К.: НУХТ, 2013. – 174 с.
2. Управління ІТ проектами [Електронний ресурс]: лабораторний практикум для студентів напряму підготовки 6.050101 "Комп'ютерні науки" денної та заочної форм навч. / уклад. О. А. Хлобистова, М. В. Гладка. - К.: НУХТ, 2013. – 108 с. - Режим доступу: <http://library.nuft.edu.ua/ebook/file/51.07A.pdf>
3. Управління ІТ проектами [Електронний ресурс]: методичні рекомендації до виконання курсової роботи для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навч. / уклад. М. В. Гладка, О. А. Хлобистова. – К. : НУХТ, 2014.– 91 с. – Режим доступу: <http://library.nuft.edu.ua/ebook/file/51.13.pdf>;
4. Технологія створення програмних продуктів [Електронний ресурс]: лабораторний практикум для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форми навч. / уклад. В.А. Литвинов, М.В. Гладка, О.А. Хлобистова. – К. : НУХТ, 2014.- 86 с. - Режим доступу: <http://library.nuft.edu.ua/ebook/file/51.17A.pdf>.

### Базова література

1. Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул. Технология разработки программного обеспечения : учебное пособие. – М.: ИД «ФОРУМ» -2008 .-400с.
2. Г.С. Иванова . Технология программирования: Учебник.- М.: Издательство МГТУ им. Н.Э. Баумана, 2002. – 243 с.
3. Ірина Бородкіна, Георгій Бородкін. Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів. Видавництво Центр навчальної літератури Мова Українська. 2018 .с 204
4. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник.- М.: Финансы и статистика, 2005.- 544 с.
- 5.А.М.Вендров. Объектно-ориентированный анализ и проектирование с использованием языка UML и Rational Rose. - Практикум

### Допоміжна література

1. В.Л.Плескач, Ю.В.Рогошина, Н.П.Кустова. Інформаційні технології та системи К.: «КНИГА», 2004.- 520 с.
2. Ф.И.Андон и др., Основы инженерии качества программных систем.- Издательский Дом «Академпериодика»,2007. – 673 с.
2. Орлов С.А. Технологии разработки программного обеспечения: Учебник. Санкт-Петербург, ООО «Питер Принт», 2002. – 316 с.
3. Н.В.Красильникова, Проектирование информационных систем: Учебное пособие. –М.: МИСиС, 2004.-106 с.

Програму склав  
д.т.н. проф. **ЛИТВИНОВ В.А.**