

| | |
|---|--|
|  | <p style="text-align: center;">СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «ТЕХНОЛОГІЯ СТВОРЕННЯ ПРОГРАМНИХ ПРОДУКТІВ»</p> <p>Рівень вищої освіти: Перший (бакалаврський)</p> <p>Спеціальність: <u>F3 Комп'ютерні науки</u></p> <p>Рік навчання: <u>4 -й, 7 семестр</u></p> <p>Кількість кредитів ECTS: <u>4 кредитів</u></p> <p>Назва кафедри: <u>Комп'ютерних наук та</u> <u>цифрової економіки</u></p> <p>Мова викладання: <u>українська</u></p> |
| Лектор курсу | к.т.н., ст. викл. Суприган Віталій Анатолійович |
| Контактна інформація лектора (e-mail) | suprigan@gmail.com |

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна «Технологія створення програмних продуктів» є обов'язковою компонентою ОПП.

Загальний обсяг дисципліни 120 год.: лекції – 26 год.; практичні заняття – 24 год., самостійна робота – 70 год.

Формат проведення: лекції, практичні заняття, консультації.

Підсумковий контроль – екзамен

Під час вивчення дисципліни можуть використовувати знання, отримані з таких дисциплін: «Програмування», «Організація баз даних та знань», «Контроль якості програмного забезпечення», «Об'єктно-орієнтоване програмування», «Технології розподілених систем та паралельних обчислень», «Хмарні технології», «Проектування інформаційних систем».

Основні положення навчальної дисципліни можуть застосовуватися під час вивчення таких дисциплін: «Програмування мобільних додатків», Міждисциплінарна курсова робота, Виробнича практика.

Призначення навчальної дисципліни

Дисципліна «Технологія створення програмних продуктів» призначена для формування у фахівців розуміння етапів, вимог, послідовності технологічних етапів процесу створення та підтримки програмного забезпечення. Зокрема, посилення критичного мислення, управлінських, логічних та професійних компетентностей, які дозволяють б працювати у сучасних професійних умовах, зокрема, в галузі АПК.

Мета вивчення навчальної дисципліни

Метою вивчення навчальної дисципліни «Технологія створення програмних продуктів» є формування у студентів знань, вмінь та професійних навичок необхідних для побудови технологічних послідовностей та забезпечення процесів побудови сучасного програмного забезпечення, дотримання стандартних та загальноприйнятих методології функціонування процесів у інформаційних у різних галузях господарювання, зокрема у АПК.

У підсумку, студенти мають володіти основними підходами до організації розробки методології створення програмних систем, організації потрібних завдань на різних фазах розробки програмної системи.

Завдання вивчення дисципліни

Навчити майбутніх спеціалістів володіти основами ІТ-технологій та інструментами апарату для створення технологій програмного забезпечення, програмних засобів. Навчитися використовувати методи управління проектами та набути вміння застосовувати інструменти та методології управління ІТ-проектами в АПК із застосуванням інформаційних технологій; набути теоретичні та практичні навички, а також самостійного засвоєння теоретичних, методичних основ формування вивчення основних принципів та прийомів з питань технології та управління діяльністю в сфері АПК; набуття знань як вихідної основи для формування навичок і вмінь дослідження практичних проблем у цій сфері. Виховувати у здобувачів вищої освіти визначеність і наполегливість щодо поставлених завдань і взятих обов'язків.

Після проходження дисципліни студенти будуть вміти аналізувати предметну область, будувати діаграми з описом програмної системи. Знати етапи розробки програмної системи. Вміти вибрати метрики оцінки вимог та якості системи.

ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ, ЯКИХ НАБУВАЄ ЗДОБУВАЧ ПРИ ВИВЧЕНІ ДИСЦИПЛІНИ ВІДПОВІДНО ДО ОСВІТНЬОЇ ПРОГРАМИ

У результаті вивчення навчальної дисципліни здобувач повинен сформулювати такі програмні компетентності:

інтегральну компетентність (ІК):

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

загальні компетентності (ЗК):

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК9. Здатність працювати в команді.

ЗК11. Здатність приймати обґрунтовані рішення.

спеціальні (фахові) компетентності (СК):

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

СК13. Здатність до розробки мережевого програмного забезпечення, що функціонує на основі різних топологій структурованих кабельних систем, використовує комп'ютерні системи і мережі передачі даних та аналізує якість роботи комп'ютерних мереж.

СК16. Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації.

ПРОГРАМНІ РЕЗУЛЬТАТИ НАВЧАННЯ ВІДПОВІДНО ДО ОСВІТНЬОЇ ПРОГРАМИ

РН5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

РН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

РН11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію

(техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

PH13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

PH16. Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних.

Вивчення даної дисципліни формує у здобувачів освіти соціальні навички (soft skills): комунікативність (реалізується через: метод роботи в парах та групах, робота з інформаційними джерелами), робота в команді (реалізується через: метод проектів), лідерські навички (реалізується через: робота в групах, метод проектів, метод самопрезентації).

ПЛАН ВИВЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

| № з/п | Назви теми | Форми організації навчання та кількість годин | | Самостійна робота, кількість годин |
|--------------|---|---|-------------------|------------------------------------|
| | | лекційні заняття | практичні заняття | |
| 1 | Тема 1. Програмний продукт і процес його створення. Ролі в команді та артефакти розробки. Від постановки задачі до підтримки. | 2 | | 6 |
| 2 | Тема 2. Вимоги та моделювання предметної області. Сценарії використання і user stories. Формалізація вимог і критерії приймання. Прототипування як інструмент уточнення. | 2 | 2 | 6 |
| 3 | Тема 3. Проектування рішень. Декомпозиція на модулі та компоненти. Інтерфейси і контракти. Трасування вимог до реалізації. | 2 | 2 | 6 |
| 4 | Тема 4. Архітектура клієнт сервер і сервісна архітектура. Проектування API. Інтеграція сервісів і робота з чергами подій. Патерни взаємодії в розподілених системах. | 2 | 2 | 6 |
| 5 | Тема 5. Дані в програмних продуктах. Проектування схеми даних і доступу до даних. Узгодженість, транзакції, конкурентність. Міграції та версіонування даних. | 2 | 2 | 6 |
| 6 | Тема 6. Парадигми програмування на практиці. Вибір парадигми під задачу і обмеження. Узагальнене, об'єктне, функціональне та реактивне програмування. Компроміси якості коду. | 2 | 2 | 5 |
| 7 | Тема 7. Принципи проектування коду і повторне використання. SOLID та інші принципи підтримованості. Патерни проектування. Антипатерни і технічний борг. | 2 | 2 | 5 |
| 8 | Тема 8. Керування версіями і командна робота. Стратегії гілкування і релізні потоки. Code review як контроль якості. Управління змінами і вирішення конфліктів. | 2 | 2 | 5 |
| 9 | Тема 9. Тестування і стратегія якості продукту. Піраміда тестів і тестові рівні. Контрактне тестування і тестування інтеграцій. Нефункціональні перевірки. Спостережуваність через логи, метрики, трасування. | 2 | 2 | 5 |
| 10 | Тема 10. CI/CD і автоматизація життєвого циклу. Збірка, статичний аналіз, перевірки безпеки. Пакування і артефакти. Релізи, відкат, feature flags. Автоматизація експлуатаційних перевірок. | 2 | 2 | 5 |
| 11 | Тема 11. Контейнеризація і розгортання в хмарі. Docker і базові практики контейнерів. Оркестрація і середовища виконання в хмарі. Конфігурації та секрети. Надійність і масштабування. | 2 | 2 | 5 |
| 12 | Тема 12. Мережеві аспекти програмних продуктів. Протоколи прикладного рівня і мережні помилки. Балансування і маршрутизація. Безпечна взаємодія сервісів. Аналіз якості мережевої роботи і діагностика. | 2 | 2 | 5 |
| 13 | Тема 13. Безпека і стійкість програмного продукту. Безпечне проектування і моделювання загроз. Керування уразливостями і залежностями. Контроль доступу і захист даних. Планування відмовостійкості, резервування, відновлення після збоїв. Оцінювання рішень в умовах невизначеності і ризиків. | 2 | 2 | 5 |
| Разом | | 26 | 24 | 70 |

Самостійна робота здобувача вищої освіти

Самостійна робота здобувача організується шляхом видачі індивідуального переліку питань і практичних завдань з кожної теми, які не виносяться на аудиторне опрацювання та виконання індивідуального творчого завдання.

Самостійна робота здобувача є одним із способів активного, цілеспрямованого набуття нових для нього знань та умінь. Вона є основою його підготовки як фахівця, забезпечує набуття ним прийомів пізнавальної діяльності, інтерес до творчої роботи, здатність вирішувати наукові та практичні завдання.

Виконання здобувачем самостійної роботи передбачає, за необхідності, отримання консультацій або допомоги відповідного фахівця. Навчальний матеріал навчальної дисципліни, передбачений робочою програмою для засвоєння здобувачем у процесі самостійної роботи, виносяться на поточний і підсумковий контроль поряд з навчальним матеріалом, який опрацьовувався під час аудиторних занять. Організація самостійної роботи здобувачів передбачає: планування обсягу, змісту, завдань, форм і методів контролю самостійної роботи, розробку навчально-методичного забезпечення; виконання здобувачем запланованої самостійної роботи; контроль та оцінювання результатів, їх систематизацію, оцінювання ефективності виконання здобувачем самостійної роботи.

Індивідуальні завдання здобувач виконує самостійно під керівництвом викладача згідно з індивідуальним навчальним планом.

У випадку реалізації індивідуальної освітньої траєкторії здобувача заняття можуть проводитись за індивідуальним графіком.

Види самостійної роботи

| № п/п | Вид самостійної роботи | Години | Термін виконання | Форма та метод контролю |
|--------------|---|-----------|------------------|-----------------------------|
| 1 | Підготовка до лекційних занять та підготовка презентаційних матеріалів | 26 | щотижнево | Усне та письмове опитування |
| 2 | Підготовка до практичних занять та підготовка презентаційних матеріалів | 12 | щотижнево | Усне та письмове опитування |
| 3 | Підготовка самостійних питань з тематики дисципліни | 20 | щотижнево | Усне та письмове опитування |
| 4 | Підготовка до контрольних робіт та тестування | 12 | 1 раз на 2 тижні | Тестування у системі Moodle |
| Разом | | 70 | | |

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Основна

1. Операційні системи: навч. посіб. / [І. М. Федотова-Півень, І. В. Миронець, О. Б. Півень, С. В. Сисоєнко, Т. В. Миронюк]; за ред. В. М. Рудницького. Харків: ТОВ «ДІСА ПЛЮС», 2019. 216 с.
2. Хмарні технології: навч. посіб. / [О.В. Зінченко, С.М. Іщеряков, С.В. Прокопов, С.О. Сєрих, В.В. Василенко]. К: ФОП Гуляєва В.М., 2020. 74 с.
3. Николайчук Я. М., Возна Н. Я., Пітух І. Р. Проектування спеціалізованих комп'ютерних систем : навч. Посібник. Тернопіль : ТзОВ «Терно-граф», 2010. 392 с.
4. Технології програмування та створення програмних продуктів конспект лекцій для студ. напряму підготовки 6.050101 "Комп'ютерні науки" усіх форм навчання / О. В. Алексєнко. Суми : СумДУ, 2013. 133 с.
5. Карпенко М. Ю., Манакова Н. О., Гавриленко І. О. Технології створення програмних продуктів та інформаційних систем : навч. Посібник; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекєтова. Харків : ХНУМГ ім. О. М. Бекєтова, 2017. 93 с.
6. Gayathri Mohan Full Stack Testing. A Practical Guide for Delivering High Quality Software. O'reilly Media, 2022. 406 p.

Додаткові

1. Arnon Axelrod Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects 1st ed. Edition. Print2print, 2018. 588 p.
2. Brian Okken Python Testing with pytest. Simple, Rapid, Effective, and Scalable. 2nd Edition. Pragmatic Bookshelf, 2022. 274 p.
3. Myron Marston, Ian Dees Effective Testing withRSpec 3: Build Ruby Apps with Confidence 1st Edition. Pragmatic Bookshelf, 2017. 356 p.
4. Andrew Pollner, Mark Fewster, Ina Schieferdecker Test Automation Engineer: Guide to the ISTQB Advanced Level Certification. Rocky Nook, 2018. 300 p.
5. Shen J. J. Software Testing. Techniques, Principles, and Practices. Paperback. Independently published, 2019. 286 p.
6. Rex Black, Dorothy Graham, Erik van Veenendaal Foundations of Software Testing ISTQB Certification, 4th edition. Cengage Learning EMEA, 2020. 288 p.
7. Lucas da Costa Testing JavaScript Applications. Manning Publications Co., 2021. 512 p.
8. Brian Okken Python Testing with pytest. Simple, Rapid, Effective, and Scalable. 2nd Edition. Pragmatic Bookshelf, 2022. 274 p.
9. Подчашинський Ю.О. Проектування комп'ютеризованих систем управління технологічними процесами : навч. посіб.; Житомир. держ. технол. ун-т. Житомир: ЖДТУ, 2018. 199 с.
10. Приймак В.М. Управління проектами: навч. посіб. Київ: КНУ імені Тараса Шевченка, 2017. 459 с.

Інформаційні ресурси в Інтернеті

1. Офіційний сайт компанії JetBrains [Електронний ресурс] / JetBrains.
Режим доступу: <https://www.jetbrains.com>
2. Офіційна сторінка Python: <https://www.python.org>
3. Офіційна сторінка Amazon WS: <https://aws.amazon.com>
4. Офіційна сторінка Google Cloud Platform: <https://cloud.google.com>
5. Методичні розробки (внутрішній сайт ВНАУ).
6. Тестові завдання з дисципліни (внутрішній сайт ВНАУ).

СИСТЕМА ОЦІНЮВАННЯ ТА ВИМОГИ ДО КОНТРОЛЮ ЗНАНЬ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ

У кінці семестру, здобувач вищої освіти може набрати до 60% підсумкової оцінки за виконання всіх видів робіт, що виконуються протягом семестру, до 10% за показники наукової, інноваційної, навчальної, виховної роботи та студентської активності і до 30% підсумкової оцінки – за результатами підсумкового контролю.

| | Вид навчальної діяльності | Бали |
|--------------------|--|------------|
| Атестація 1 | | |
| 1 | Участь у дискусіях на лекційних заняттях | 7 |
| 3 | Виконання і захист практичної роботи | 18 |
| 4 | Виконання самостійних завдань за тематикою дисципліни | 2 |
| 5 | Тестування | 3 |
| | Всього за атестацію 1 | 30 |
| Атестація 2 | | |
| 6 | Участь у дискусіях на лекційних заняттях | 6 |
| 8 | Виконання і захист практичної роботи | 18 |
| 9 | Виконання самостійних завдань за тематикою дисципліни | 3 |
| 10 | Тестування | 3 |
| | Всього за атестацію 2 | 30 |
| | Показники наукової, інноваційної, навчальної, виховної роботи та студентської активності | 10 |
| | Підсумкове тестування | 30 |
| | Разом | 100 |

Якщо здобувач упродовж семестру за підсумками контрольних заходів набрав менше 35 балів, то він не допускається до заліку чи екзамену. Крім того, обов'язковим при мінімальній кількості балів за підсумками контрольних заходів є виконання індивідуальної творчої роботи (презентації).

Під час виконання навчальних завдань, а також завдань поточних та підсумкових контрольних заходів не допустимо порушення академічної доброчесності. Презентації та виступи мають бути авторськими та оригінальними, інформація про результати власної навчальної (наукової, творчої) діяльності – достовірною; у разі використання ідей, розробок, тверджень, відомостей мають бути посилання на джерела інформації з

дотриманням норм законодавства про авторське право і суміжні права.

Програма навчальної дисципліни передбачає врахування результатів неформальної та інформальної освіти при наявності підтверджуючих документів як окремі кредити вивчення навчальних дисциплін.

Переведення балів внутрішньої 100-бальної шкали в національну здійснюється у відповідності до шкали.

Відповідність шкал оцінок якості засвоєння навчального матеріалу

| Сума балів за всі види навчальної діяльності | Оцінка ECTS | Оцінка за національною шкалою для екзамену |
|--|-------------|--|
| 90 – 100 | A | відмінно |
| 82-89 | B | добре |
| 75-81 | C | |
| 66-74 | D | |
| 60-65 | E | задовільно |
| 35-59 | FX | незадовільно з можливістю повторного складання |
| 0-34 | F | незадовільно з обов'язковим повторним вивченням дисципліни |