



Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»



Кафедра інженерії
програмного
забезпечення в енергетиці
(ІПЗЕ)

ПРОГРАМНІ ФРЕЙМВОРКИ ДЛЯ ІНТЕЛЕКТУАЛЬНИХ РОЗПОДІЛЕНИХ АВТОНОМНИХ АГЕНТІВ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 «Інформаційні технології»</i>
Спеціальність	<i>121 «Інженерія програмного забезпечення»</i>
Освітня програма	<i>Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>4 курс, осінній семестр</i>
Обсяг дисципліни	<i>120 годин / 4 кредити ЕКТС (лекції – 36 год., практичні заняття – 18 год., самостійна робота – 66 год.)</i>
Семестровий контроль/ контрольні заходи	<i>Залік / захист практичних занять, модульні тести, підсумковий тест або захист міні-проєкту</i>
Розклад занять	<i>http://rozklad.kpi.ua</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор:ст. викл. Сарибіга Ганна Володимирівна</i>
Розміщення курсу	<i>(LMS кафедри / репозиторій — заповнюється кафедрою)</i>

Програма навчальної дисципліни

Дисципліна розглядає принципи та інженерні підходи до створення програмного забезпечення для роботизованих систем — складних кіберфізичних систем, що поєднують сенсорні підсистеми, виконавчі механізми, алгоритми керування та програмні компоненти для обробки даних і прийняття рішень. У межах курсу розглядаються сучасні програмні фреймворки, що використовуються для розробки роботів і автономних систем, зокрема фреймворки для інтеграції сенсорів, керування приводами, обробки даних та координації роботи програмних модулів.

Курс орієнтований на інженерний підхід до розробки програмного забезпечення для робототехнічних систем і охоплює як фундаментальні принципи архітектури роботів, так і практичні аспекти створення програмних компонентів для мобільних роботів, роботизованих маніпуляторів та автономних пристроїв. Особлива увага приділяється модульній архітектурі програмного забезпечення роботів, взаємодії програмних компонентів, використанню middleware-рішень та інструментів для розробки складних робототехнічних систем.

Навчальна дисципліна реалізується у форматі дуальної освіти у співпраці з ІТ-компанією **N-iX**, що дозволяє поєднати академічну підготовку студентів із практичними підходами,

які використовуються у сучасних інженерних командах, що розробляють програмне забезпечення для роботизованих систем, автономних платформ, IoT-пристроїв та кіберфізичних систем.

У межах курсу студенти знайомляться з сучасними програмними фреймворками для робототехніки, зокрема з **Robot Operating System (ROS/ROS2)**, що широко використовується для створення роботизованих систем у наукових та промислових проєктах. Розглядаються принципи організації програмних компонентів роботів, обміну повідомленнями між модулями, інтеграції сенсорних систем, управління приводами та реалізації алгоритмів навігації.

Курс охоплює архітектуру програмного забезпечення роботів, принципи побудови розподілених робототехнічних систем, роботу з ROS-вузлами (nodes), повідомленнями та сервісами, організацію програмних пакетів, використання бібліотек робототехніки, а також інтеграцію сенсорів (камери, лідари, інерціальні датчики, ультразвукові сенсори) та виконавчих механізмів (електродвигуни, сервоприводи, маніпулятори).

Окремий блок дисципліни присвячений інтеграції програмного забезпечення роботів з алгоритмами навігації та керування рухом, обробці даних сенсорів, роботі з системами локалізації та побудови карт середовища. Розглядаються інструменти візуалізації робототехнічних систем, симулятори роботів та засоби тестування програмного забезпечення роботизованих платформ.

З огляду на дуальний формат навчання значна увага приділяється практичним завданням, що відображають реальні процеси розробки програмного забезпечення для робототехнічних систем у сучасних IT-компаніях. Практичні заняття передбачають створення програмних модулів роботів, інтеграцію сенсорів і виконавчих механізмів, а також розробку програмного забезпечення для роботизованих платформ із використанням сучасних фреймворків і бібліотек.

Метою навчальної дисципліни є формування та закріплення у студентів наступних компетентностей: здатність до абстрактного мислення, аналізу та синтезу; здатність застосовувати знання у практичних ситуаціях; здатність вчитися і оволодівати сучасними знаннями; здатність до пошуку, оброблення та аналізу інформації з різних джерел; здатність працювати в команді; здатність аналізувати архітектуру роботизованих систем та їх програмні компоненти; здатність розробляти програмне забезпечення для роботизованих платформ; здатність проектувати програмні модулі роботів та інтегрувати їх у єдину систему; здатність забезпечувати якість програмного забезпечення для робототехнічних систем; здатність використовувати сучасні фреймворки та бібліотеки для розробки роботизованих систем; здатність реалізовувати програмні компоненти для керування роботизованими системами; здатність інтегрувати програмне забезпечення з сенсорними та виконавчими підсистемами роботів; здатність обґрунтовано обирати інструментарій для розробки програмного забезпечення робототехнічних систем.

Зокрема, дисципліна спрямована на формування навичок створення програмної архітектури роботизованих систем, організації взаємодії між програмними модулями, інтеграції сенсорів і виконавчих механізмів, використання фреймворків для розробки роботів та застосування сучасних інструментів тестування і налагодження робототехнічного програмного забезпечення.

Предмет навчальної дисципліни — програмні фреймворки та інструменти розробки роботизованих систем: архітектура програмного забезпечення роботів; використання ROS та інших middleware-рішень; організація взаємодії програмних компонентів; робота з сенсорними системами; керування приводами та роботизованими маніпуляторами; обробка даних сенсорів; інтеграція алгоритмів навігації; тестування та налагодження робототехнічного програмного забезпечення.

Програмні результати навчання, на формування та покращення яких спрямована дисципліна: знати принципи побудови роботизованих систем та їх програмної архітектури; застосовувати сучасні фреймворки та інструменти розробки робототехнічного програмного забезпечення; використовувати методи інтеграції програмних компонентів у роботизованих системах; застосовувати підходи до проектування програмного забезпечення для автономних роботів; використовувати інструменти розробки, тестування та налагодження робототехнічного програмного забезпечення; мати навички командної розробки програмного забезпечення для роботів; застосовувати методи тестування та перевірки робототехнічних програмних систем; забезпечувати якість програмного забезпечення для роботизованих систем; використовувати інструменти керування програмними проєктами; документувати результати розробки робототехнічного програмного забезпечення та презентувати створені системи.

Після завершення дисципліни студент уміє: використовувати сучасні програмні фреймворки для створення роботизованих систем; розробляти програмні компоненти для роботів; працювати з ROS-вузлами, повідомленнями та сервісами; інтегрувати сенсори та виконавчі механізми у робототехнічну систему; реалізовувати програмне керування рухом роботів; обробляти дані сенсорів та використовувати їх у системах керування; використовувати інструменти налагодження та тестування робототехнічного програмного забезпечення; проектувати програмну архітектуру роботизованих систем та забезпечувати надійне функціонування робототехнічних платформ.

Пререквізити та постреквізити дисципліни

Пререквізити: *основи програмування; алгоритми та структури даних; програмування мікроконтролерів; основи кіберфізичних систем; архітектура комп'ютерних систем.*

Постреквізити: *курсіві та дипломні проєкти з робототехніки та кіберфізичних систем; дисципліни з автономних робототехнічних систем, інтелектуальних систем керування, комп'ютерного зору; індустріальні проєкти з робототехніки, IoT та автономних систем; практика та стажування у компаніях, що розробляють робототехнічні та кіберфізичні системи.*

1. Зміст навчальної дисципліни

Розділ 1. Архітектура роботизованих систем та роль програмних фреймворків

Тема 1.1. Контекст роботизованих систем у сучасних технічних рішеннях та вимоги до робототехнічного ПЗ

Тема 1.2. Архітектура роботизованих систем: сенсори, приводи, контролери, обчислювальні модулі

Тема 1.3. Програмні фреймворки та middleware у робототехніці: призначення, підходи, порівняння

Тема 1.4. Компонентна та модульна архітектура ПЗ роботів: вузли, модулі, інтерфейси, контракти

Тема 1.5. Життєвий цикл робототехнічного ПЗ: інтеграція компонентів, тестування, реліз та супровід

Розділ 2. ROS/ROS2 як базовий фреймворк роботизованих систем

Тема 2.1. Основні поняття ROS/ROS2: nodes, topics, services, actions, parameters

Тема 2.2. Модель обміну даними: повідомлення, QoS (для ROS2), частота та затримки

Тема 2.3. Пакети та залежності: структура workspace, build system (colcon/catkin), запуск та конфігурація

Тема 2.4. Інструменти ROS: rosbag, rqt, RViz, launch-системи, діагностика і логування

Тема 2.5. Трансформації координат і часові узгодження: TF/TF2, timestamps, синхронізація потоків даних

Розділ 3. Інтеграція сенсорів та виконавчих механізмів у роботизованих системах

Тема 3.1. Сенсорні підсистеми робота: камери, IMU, lidar/sonar, енкодери — типові інтерфейси і драйвери

Тема 3.2. Обробка та фільтрація даних сенсорів у фреймворках робототехніки

Тема 3.3. Керування приводами: двигуни, сервоприводи, контролери, PID-регулятори, профілі руху

Тема 3.4. Роботизовані маніпулятори: кінематика, планування траєкторій, інтеграція керування

Тема 3.5. Апаратна інтеграція: взаємодія з мікроконтролерами (serial/CAN/ethernet), протоколи та шлюзи

Розділ 4. Навігація, локалізація та автономна поведінка

Тема 4.1. Локалізація та оцінювання стану робота: одометрія, IMU, sensor fusion (огляд)

Тема 4.2. Побудова карти та робота з середовищем: основи SLAM (огляд), формати карт

Тема 4.3. Планування руху та уникнення перешкод: глобальне/локальне планування (огляд)

Тема 4.4. Архітектура поведінки робота: state machine, behavior tree (на рівні інтеграції компонентів)

Розділ 5. Інженерні практики розробки робототехнічного ПЗ

Тема 5.1. Toolchain та збірка робототехнічного ПЗ: залежності, контейнеризація (Docker), відтворені збірки

Тема 5.2. Налаштування та діагностика: логування, профілювання, аналіз затримок, робота з багами інтеграції

Тема 5.3. Тестування та забезпечення якості: unit/integration tests, симуляція, quality gates, CI для robotics

Тема 5.4. Документування та супровід: API-документація, конфігурації, експлуатаційні інструкції, релізи

Розділ 6. Індустріальні кейси та командні проекти (дуальна освіта з N-іХ)

Тема 6.1. Інтеграційні задачі робототехнічного ПЗ у кіберфізичних системах: сенсори → обробка → керування

Тема 6.2. Розробка модулів роботизованих систем для мобільних платформ та маніпуляторів

Тема 6.3. Командна розробка робототехнічного проекту у співпраці з компанією N-іХ

2. Навчальні матеріали та ресурси

Основна література

1. Quigley M., Gerkey B., Smart W. *Programming Robots with ROS: A Practical Introduction to the Robot Operating System* : навчальний посібник / Morgan Quigley, Brian Gerkey, William D. Smart. – Sebastopol : O'Reilly Media, 2015. – 448 с.
2. Cousins S., Foote T., Gerkey B. *ROS Robotics Projects* / Samer Cousins, Tully Foote, Brian Gerkey. – Birmingham : Packt Publishing, 2017. – 350 с.
3. Siegwart R., Nourbakhsh I., Scaramuzza D. *Introduction to Autonomous Mobile Robots* / Roland Siegwart, Illah Nourbakhsh, Davide Scaramuzza. – 2nd ed. – Cambridge : MIT Press, 2011. – 520 с.
4. Corke P. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB and Python* / Peter Corke. – 2nd ed. – Cham : Springer, 2017. – 693 с.
5. Thrun S., Burgard W., Fox D. *Probabilistic Robotics* / Sebastian Thrun, Wolfram Burgard, Dieter Fox. – Cambridge : MIT Press, 2005. – 647 с.
6. Niku S. *Introduction to Robotics: Analysis, Control, Applications* / Saeed B. Niku. – 2nd ed. – Hoboken : Wiley, 2010. – 512 с.
7. Open Robotics. *ROS2 Documentation* : офіційна технічна документація. – Режим доступу: <https://docs.ros.org>
8. Gazebo Simulation Environment Documentation. – Режим доступу: <https://gazebo.org>

Допоміжна література

1. Corke P. *Robotics Toolbox for Python*. – Springer, 2023.
2. Macenski S., Foote T., Gerkey B. *Robot Operating System 2 (ROS2): Design and Architecture*. – Open Robotics Documentation.
3. Beard R., McLain T. *Small Unmanned Aircraft: Theory and Practice*. – Princeton University Press, 2012.
4. LaValle S. *Planning Algorithms*. – Cambridge University Press, 2006.
5. OpenCV Organization. *OpenCV Documentation* – <https://docs.opencv.org>

Інформаційні ресурси

1. Електронний ресурс: Software Engineering 2014 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE Computer Society.
2. Електронний ресурс: Guide to the Software Engineering Body of Knowledge (SWEBOK®) Version 3.0. IEEE Computer Society.
3. Інтранет-сайт кафедри інженерії програмного забезпечення вбудованих систем [Електронний ресурс] – <http://ipze.kpi.ua/metodichki>
4. Модульне середовище університету eCampus КПІ ім. Ігоря Сікорського [Електронний ресурс] – <https://ecampus.kpi.ua>
5. Офіційна документація ROS [Електронний ресурс] – <https://docs.ros.org>
6. Офіційна документація ROS2 [Електронний ресурс] – <https://docs.ros.org/en/foxy/index.html>
7. Документація симулятора роботів Gazebo [Електронний ресурс] – <https://gazebo.org>
8. OpenCV Documentation [Електронний ресурс] – <https://docs.opencv.org>
9. Репозиторії робототехнічних бібліотек та прикладів ROS-проектів [Електронний ресурс] – <https://github.com>

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань
1	Лекція 1. Тема 1.1. Контекст роботизованих систем. Основні питання: поняття робототехнічних систем; області застосування (промисловість, логістика, автономні транспортні системи, сервісні роботи); роль програмного забезпечення в робототехніці; вимоги до надійності та реального часу.
2	Лекція 2. Тема 1.2. Архітектура роботизованих систем. Основні питання: сенсорні системи; виконавчі механізми; контролери та обчислювальні модулі; структура програмного забезпечення роботів.
3	Лекція 3. Тема 1.3. Програмні фреймворки в робототехніці. Основні питання: роль middleware; порівняння ROS, ROS2 та інших платформ; принципи модульної архітектури робототехнічного ПЗ.

4	Лекція 4. Тема 1.4. Компонентна архітектура програмного забезпечення роботів. Основні питання: програмні модулі; взаємодія компонентів; програмні інтерфейси; принципи повторного використання коду.
5	Лекція 5. Тема 1.5. Життєвий цикл програмного забезпечення роботів. Основні питання: проектування системи; інтеграція компонентів; тестування робототехнічних систем; підтримка та розвиток програмного забезпечення.
6	Лекція 6. Тема 2.1. Основи ROS/ROS2. Основні питання: nodes; topics; services; actions; параметри; структура ROS-системи.
7	Лекція 7. Тема 2.2. Обмін повідомленнями у ROS. Основні питання: типи повідомлень; асинхронний обмін даними; QoS у ROS2; часові обмеження робототехнічних систем.
8	Лекція 8. Тема 2.3. Структура ROS-пакетів. Основні питання: workspace; build-система; залежності пакетів; запуск ROS-вузлів.
9	Лекція 9. Тема 2.4. Інструменти ROS. Основні питання: rosbag; rqt; RViz; launch-файли; моніторинг системи.
10	Лекція 10. Тема 2.5. Координатні системи та трансформації. Основні питання: TF/TF2; системи координат робота; синхронізація сенсорних даних.
11	Лекція 11. Тема 3.1. Сенсорні системи роботів. Основні питання: типи сенсорів; інтеграція сенсорів; драйвери пристроїв.
12	Лекція 12. Тема 3.2. Обробка сенсорних даних. Основні питання: фільтрація сигналів; обробка потоків даних; використання бібліотек обробки сенсорної інформації.
13	Лекція 13. Тема 3.3. Керування приводами роботів. Основні питання: двигуни; драйвери моторів; PID-регулювання; керування рухом.

14	Лекція 14. Тема 3.4. Роботизовані маніпулятори. Основні питання: кінематика; планування траєкторій; програмне керування маніпуляторами.
15	Лекція 15. Тема 4.1. Локалізація та навігація роботів. Основні питання: одометрія; <i>sensor fusion</i> ; основи локалізації.
16	Лекція 16. Тема 4.2. Планування руху роботів. Основні питання: алгоритми планування; уникнення перешкод; навігаційні системи роботів.
17	Лекція 17. Тема 5.1. Інструменти розробки робототехнічного ПЗ. Основні питання: системи збірки; налагодження програм; тестування робототехнічних систем.
18	Лекція 18. Тема 6.1. Індустріальні робототехнічні проєкти. Основні питання: архітектура реальних робототехнічних систем; командна розробка; інтеграція програмного забезпечення роботів у промислових системах.

Практичні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	Практичне заняття №1. Налаштування середовища ROS. (пов'язано з лекціями 1–3). Основні питання: встановлення ROS/ROS2; створення <i>workspace</i> ; запуск перших вузлів; структура ROS-проєкту.
2	Практичне заняття №2. Створення ROS-вузлів. (пов'язано з лекцією 6). Основні питання: створення <i>node</i> ; обмін повідомленнями; використання <i>topics</i> .
3	Практичне заняття №3. Робота з повідомленнями та сервісами. (пов'язано з лекціями 6–7). Основні питання: створення типів повідомлень; використання <i>services</i> ; передача даних між вузлами.

4	Практичне заняття №4. Використання інструментів ROS. (пов'язано з лекцією 9). Основні питання: <i>roslaunch; rqt; RViz; візуалізація даних сенсорів.</i>
5	Практичне заняття №5. Інтеграція сенсорів. (пов'язано з лекцією 11). Основні питання: <i>підключення сенсорів; отримання та обробка даних.</i>
6	Практичне заняття №6. Керування приводами. (пов'язано з лекцією 13). Основні питання: <i>керування двигунами; створення контролерів руху.</i>
7	Практичне заняття №7. Робота з симулятором роботів. (пов'язано з лекцією 15). Основні питання: <i>запуск симулятора Gazebo; моделювання роботизованої системи.</i>
8	Практичне заняття №8. Налаштування та тестування робототехнічного ПЗ. (пов'язано з лекцією 17). Основні питання: <i>аналіз помилок; тестування ROS-модулів.</i>
9	Практичне заняття №9. Командний проєкт роботизованої системи. (пов'язано з лекцією 18). Основні питання: <i>інтеграція програмних модулів; створення робототехнічної системи; демонстрація роботи; захист командного проєкту у співпраці з компанією N-iX.</i>

Самостійна робота студента

<i>№ з/п</i>	<i>Вид самостійної роботи</i>	<i>Кількість годин СРС</i>
1	<i>Опрацювання технічної документації програмних фреймворків для робототехніки (ROS/ROS2), інструментів розробки та бібліотек робототехнічного програмного забезпечення</i>	8

2	<i>Вивчення архітектури роботизованих систем: сенсорні підсистеми, виконавчі механізми, контролери, програмні модулі</i>	8
3	<i>Самостійне опрацювання матеріалів щодо створення ROS-вузлів, повідомлень, сервісів та взаємодії між програмними компонентами</i>	10
4	<i>Підготовка до практичних занять (аналіз завдань, розробка та тестування програмних модулів роботизованої системи)</i>	12
5	<i>Розробка програмних модулів для командного робототехнічного проєкту (створення ROS-пакетів, інтеграція сенсорів та приводів, реалізація алгоритмів керування)</i>	18
6	<i>Тестування, налагодження та підготовка демонстрації роботи роботизованої системи (симуляція або робота на реальній платформі)</i>	6
7	<i>Підготовка до модульного контролю та підсумкового контролю (залік / екзамен)</i>	4

5. Контрольні заходи

Поточний контроль: захист практичних занять; перевірка репозиторію та project board; модульні тести за темами.

Семестровий контроль: залік у формі підсумкового тесту та/або захисту міні-проєкту (визначається викладачем).

Політика та контроль

3. Політика навчальної дисципліни (освітнього компонента)

Система вимог, які викладач ставить перед студентом:

- *правила відвідування занять: відповідно до Наказу 1-273 від 14.09.2020 р. заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або*

штрафні бали. Відповідно до PCO даної дисципліни бали нараховують за відповідні види навчальної активності на лекційних та практичних заняттях.

- правила поведінки на заняттях: студент має можливість отримувати бали за відповідні види навчальної активності на лекційних та практичних заняттях, передбачені PCO дисципліни. Використання засобів зв'язку для пошуку інформації на гугл-диску викладача, в інтернеті, в дистанційному курсі на платформі Сікорський здійснюється за умови вказівки викладача;
- політика дедлайнів та перескладань: якщо студент не проходив або не з'явиться на модульний тест (без поважної причини), його результат оцінюється у 0 балів. Перескладання результатів модульних тестів не передбачено.
- політика щодо академічної доброчесності: дозволено використання відкритих бібліотек/прикладів лише з коректним посиланням і поясненням власного внеску. Кодекс честі Національного технічного університету України «Київський політехнічний інститут» <https://kpi.ua/files/honorcode.pdf>.
- при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соцмережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та обмежувати спілкування робочим часом викладача.

4. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: захист практичних занять; перевірка репозиторію та project board (GitHub Projects / Jira); модульні тести.

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Умовою позитивної атестації є значення поточного рейтингу студента не менше 50 % від максимально можливого на час атестації.

Семестровий контроль: залік.

Умови допуску до семестрового контролю: семестровий рейтинг більше 30 балів.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
95-100	Відмінно
85-94	Дуже добре
75-84	Добре
65-74	Задовільно
60-64	Достатньо
Менше 60	Незадовільно
Менше 30	Не допущено

Загальна рейтингова оцінка студента після завершення семестру складається з балів, отриманих за:

- виконання та захист практичних занять;
- виконання модульних тестів (2 модульні тести);
- підсумковий контроль

Практичні заняття (9×4)	Модульні тести (2×11)	Підсумковий контроль (залік)
----------------------------	--------------------------	---------------------------------

36	22	42
----	----	----

Практичні заняття

Ваговий бал 4. Максимальна кількість балів за практичні заняття – 4 бали × 9 занять = 36 балів.

Практичні заняття здаються з демонстрацією відтворюваності: репозиторій + білд + запуск тестів + project board (GitHub Projects або Jira); для on-target частини – лог/демонстрація на стенді (за наявності). Артефакти (вимоги, моделі, дизайн-ноти, тести, реліз-ноти) мають бути версіоновані у репозиторії.

Критерії оцінювання

- завдання виконано повністю, вчасно, з відтворюваним результатом – 4 бали;
- виконано з незначними зауваженнями – 3 бали;
- виконано з суттєвими зауваженнями або із затримкою – 2 бали;
- виконано частково – 1 бал;
- не виконано – 0 балів.

Модульні тести

Ваговий бал 11. Максимальна кількість балів – 11 балів × 2 = 22 бали. Модульний тест 1 проводиться після Розділу 1 (теми 1.1–1.5: процеси, вимоги, моделювання, project management). Модульний тест 2 проводиться після Розділу 2 (теми 2.1–2.9: архітектура, тестування, CI, реліз).

Критерії оцінювання

- повна правильна відповідь – 11 балів;
- відповідь з незначними помилками – 8–10 балів;
- відповідь з суттєвими помилками – 4–7 балів;
- відповідь переважно неправильна – 1–3 бали;
- відповідь відсутня або повністю неправильна – 0 балів.

Підсумковий контроль

Форма семестрового контролю – залік. Максимальний бал – 42.

Підсумковий контроль проводиться у формі залікового тесту або захисту міні-проєкту (визначається викладачем на початку семестру).

Максимальна сума балів складає 100. Для отримання заліку «автоматом» потрібно мати рейтинг не менше 60 балів та виконані умови допуску. Студенти з рейтингом менше 60 балів або ті, хто хоче підвищити оцінку, виконують залікову контрольну роботу (при цьому набрані бали анулюються, оцінка за контрольну є остаточною).

Робочу програму навчальної дисципліни (силабус):

Складено Сарибогою Нанною Володимирівною, ст. викл. кафедри ІПЗЕ

Ухвалено кафедрою: _____ (протокол № ____ від __.__.20__)

Погоджено методичною комісією: _____ (протокол № ____ від __.__.20__)

Тематика командних проєктів (у співпраці з N-iX)

Студенти виконують командний міні-проєкт, який передбачає розробку програмного забезпечення для мікроконтролерної системи.

Можливі теми проєктів:

Система збору даних з сенсорів (температура, освітленість, рух) з передачею даних через UART або CAN.

Контролер робототехнічної платформи керування двигунами, енкодерами та сенсорами.

IoT-пристрій моніторингу середовища збір і передача телеметрії.

Система керування сервоприводами для роботизованого маніпулятора.

Мікроконтролерний контролер двигуна з використанням PWM та сенсорів.

Пристрій керування освітленням з використанням датчиків та PWM.

Мікроконтролерна система збору телеметрії для промислового обладнання із використанням RS-485 або CAN.

Контролер мобільного робота керування рухом, датчиками відстані та моторними драйверами.