



Силабус освітнього компонента

Програма навчальної дисципліни



Кросплатформене програмування

Шифр та назва спеціальності

122 – Комп'ютерні науки

Інститут

ННІ Комп'ютерного моделювання, прикладної фізики та математики

Освітня програма

Комп'ютерні науки. Моделювання, проектування та комп'ютерна графіка

Кафедра

Комп'ютерне моделювання процесів та систем (162)

Рівень освіти

Бакалавр

Тип дисципліни

Профільований пакет 1, Вибіркова

Семестр

5

Мова викладання

Українська

Викладачі, розробники



Іванченко Ксенія Вікторівна

kseniia.ivanchenko@khp.edu.ua

Кандидат технічних наук, доцент, доцент кафедри комп'ютерного моделювання процесів та систем

Автор та співавтор понад 30 наукових та методичних публікацій. Курси: "Проектування програмних систем", "Кросплатформене програмування", "Аналіз та тестування програмних систем", "Програмування мобільних пристроїв".

[Детальніше про викладача на сайті кафедри](#)

Загальна інформація

Анотація

Вивчаючи дисципліну студенти повинні оволодіти синтаксисом, ідіомами та стилем програмування на мові програмування Java; розуміти теорію типів, принципи успадкування, контролю доступу та іменування; використовувати параметризовані типи; розуміти принципи організації багатопотокового виконання методів; розуміти схеми синхронізації на основі Java-моніторів; користуватися javadoc-коментарями і анотаціями; вміти розробляти мережевий back-end на основі серверних компонентів Java.

Мета та цілі дисципліни

Мета вивчення дисципліни «Кросплатформене програмування» полягає в ознайомленні студентів з описом технологій Java і життєвого циклу продукту та вивчення мови програмування Java, використання різних конструкцій мови для створення Java додатків, використання умовних конструкцій, циклів і методів управління програмним потоком, реалізації технологій програмування Java і об'єктно-орієнтовані та кросплатформені концепції в Java програмах. Дисципліна направлена на підготовку фахівців, що володіють сучасними методами і засобами розробки алгоритмів та програм, що знають сучасну технологію програмування і уміючи застосовувати її при рішенні складних прикладних задач.

Формат занять

Лекції, лабораторні роботи, самостійна робота, консультації. Підсумковий контроль – іспит.

Компетентності

ЗК2: Здатність застосовувати знання у практичних ситуаціях.

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

ЗК7: Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

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.

СК20: Здатність проектувати та реалізовувати графічний інтерфейс користувача довільного рівня складності та сучасні застосунки з обробки та аналізу графічної інформації.

Результати навчання

ПР10: Використовувати інструментальні засоби розробки клієнт-серверних застосунків, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.

ПР11: 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).

ПР14: Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

ПР22: Застосовувати знання та навички у сфері проектування та реалізації графічного інтерфейсу користувача довільного рівня складності та сучасних застосунків з обробки та аналізу графічної інформації.

Обсяг дисципліни

Загальний обсяг дисципліни 120 год. (4 кредитів ECTS): лекції – 16 год., лабораторні роботи – 32 год., самостійна робота – 72 год.

Передумови вивчення дисципліни (пререквізити)

Дисципліна базується на знаннях та компетенціях, що набуває здобувач вищої освіти під час вивчення дисциплін: Алгоритмізація та програмування, Об'єктно-орієнтоване програмування та проектування, Програмування GUI, Організація баз даних.

Особливості дисципліни, методи та технології навчання

Лекції проводяться інтерактивно з використанням мультимедійних технологій. Для виконання лабораторних занять використовуються безкоштовні середовища розробки IntelliJ IDEA Community Edition та Eclipse IDE. На заняттях використовується проектний підхід до навчання. Навчальні матеріали доступні студентам через OneNote Class Notebook.

Програма навчальної дисципліни

Теми лекційних занять

Тема 1. Вступ

Огляд Java-технологій. Ключові концепції мови програмування Java. Особливості мови Java. Лексична структура програми. Escape послідовності unicode. Спеціальні символи. Послідовність вхідних елементів. Коментарі в тексті програми. Ідентифікатори. Ключові слова. Літерні константи (literals). Рядкові літерали і об'єкти String.

Тема 2. Основні аспекти програмування в Java

Роздільники і оператори. Типи і значення виразів. Діапазони значень. Типи значень у виразах. Втрата інформації при перетвореннях. Виклик методів. Блоки коду. Оператор вибору if. Ланцюжок if-else-if. Оператор вибору switch. Ітераційний оператор while, do – while, for. Використання міток. Послідовність виконання. Трансляція. Компіляція і інтерпретація. Основні категорії програм Java. Основні типи даних. Основні оператори для роботи з типами даних. Оголошення і ініціалізація змінних. Масив. Керуючі послідовності.

Тема 3. Об'єктно-орієнтоване програмування

Об'єктно-орієнтоване програмування. Класи і об'єкти. Методи. Конструктор класу. Інкапсуляція. Успадкування. Поліморфізм. Ключове слово super. Перевизначення методів. Пакети. Управління доступом. Модифікатор static. Модифікатор final. Використання абстрактних класів, модифікатор abstract. Інтерфейси interface. Обробка винятків Exception.

Тема 4. Колекції та потоки в Java

Огляд колекцій java.util. Основні класи ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap. Обробка рядків. Деякі методи класу String. Потоки (символьні, байтові). Серіалізація об'єктів (serialization).

Тема 5. Графічний інтерфейс користувача

Графічний інтерфейс користувача (GUI). AWT. Swing. Layout Manager. Обробка подій. Аплети. Технологія JavaFX.

Тема 6. Багатопотокове програмування

Базові і утилітарні класи Java. Пакет java.lang і базові класи. Класи Class і Object. Класи-обгортки примітивних типів. Клас Math. Класи String і StringBuffer. Багатопотокове програмування в Java. Клас Thread. Інтерфейс Runnable. Синхронізація коду. Монітори та умови. Java Memory Model.

Тема 7. Робота з базами даних

Введення в JDBC. Клас DriverManager. клас Statement.

Тема 8. Мережеві протоколи та обробка XML в Java

Модель OSI. Основні характеристики TCP і UDP. Модель «Клієнт-сервер». Інтерфейс сокетів. Клас ServerSocket. Document type definition. SAX і DOM. Робота з SAX і DOM в Java. Запис XML в Java.

Теми практичних занять

Не передбачено навчальним планом.

Теми лабораторних робіт

Лабораторна робота 1.

Ознайомлення з Java. Налаштування середовища розробки, запуск першої програми.

Лабораторна робота 2.

Розробка простого консольного застосунку з використанням масивів, рядків та файлів.

Лабораторні роботи 3-4.

Розробка консольного застосунку з використанням успадкування.

Лабораторні роботи 5-6.

Розробка консольного застосунку із застосуванням поліморфізму.

Лабораторна робота 7-8.

Розробка консольного застосунку з використанням колекцій у Java.

Лабораторна робота 8.

Обробка строк. Використання регулярних виразів у Java-додатках.

Лабораторні роботи 9-10.

Створення програми на JavaFX з використанням редактора інтерфейсу Scene Builder.

Лабораторна робота 11.

Стилізація GUI застосунку за допомогою CSS. Додавання іконки програми.

Лабораторна робота 12.

Арі-інтерфейс JavaFX canvas. Побудова базових фігур.

Лабораторні роботи 13-14.

Робота з базою даних у кросплатформеному застосунку за допомогою JDBC.

Лабораторна робота 15.

Презентація та захист індивідуальних розрахункових завдань.

Лабораторна робота 16.

Підсумкове тестування.

Самостійна робота

Курс передбачає виконання індивідуального розрахункового завдання з створення застосунку з графічним інтерфейсом користувача за обраною тематикою. Результат розробки оформлюється у презентацію, з обов'язковим захистом в кінці курсу. Оцінка виставляється за результатом відкритого захисту розрахункового завдання перед комісією, яка складається з викладача та інших студентів групи.

Студентам також рекомендуються додаткові матеріали (відео, статті) для самостійного вивчення та аналізу.

Література та навчальні матеріали

1 Java Documentation. URL: <https://docs.oracle.com/en/java/>.

2 Bloch J. Effective Java 3rd Edition, 2017.

3 Schildt H. Java: The Complete Reference, 12th Edition, 2021 - 1280 с.

4 Sierra K., Bates B. Head First Java (2nd Edition), 2005.

5 Urma R.-G., Fusco M., Mycroft A. Modern Java in Action, 2018.

6 Mak S., Bakker P. Java 9 Modularity: Patterns and Practices for Developing Maintainable Applications, 2017.

7 Gosling J. The Java® Language Specification – Java SE 8 Edition / James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. – 500 Oracle Parkway, Redwood City, California 94065, U.S.A. – 2015. – 788 p.

Система оцінювання

Критерії оцінювання успішності студента та розподіл балів

100% підсумкової оцінки складаються з результатів оцінювання у вигляді підсумкового тестування (20%) та поточного оцінювання (80%).

Підсумкове тестування: 40 тестових питань, наявні питання з декількома відповідями та відкриті питання.

Поточне оцінювання: виконання лабораторних робіт (60%) та розрахункове завдання (20%). У разі непогодження з отриманою оцінкою 100% підсумкової оцінки складаються з результатів оцінювання у вигляді екзамену. Обов'язковою умовою допуску до екзамену є виконання всіх лабораторних робіт.

Екзамен: письмове завдання (2 запитання з теорії та практичне завдання) та усна доповідь.

Шкала оцінювання

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

Норми академічної етики і політика курсу

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту. Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

Погодження

Силабус погоджено

28.08.2023

Завідувач кафедри
Дмитро БРЕСЛАВСЬКИЙ

28.08.2023

Гарант ОП
Оксана ТАТАРІНОВА