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

Кафедра обчислювальної техніки

# Основи програмування. Частина 1

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

| | |
|---|---|
| **Рівень вищої освіти** | ***Перший (бакалаврський)*** |
| **Галузь знань** | *12 Інформаційні технології* |
| **Спеціальність** | *121 Інженерія програмного забезпечення* |
| **Освітня програма** | *Інженерія програмного забезпечення комп'ютерних систем* |
| **Статус дисципліни** | *Нормативна* |
| **Форма навчання** | *очна(денна)* |
| **Рік підготовки, семестр** | *1 курс, осінній семестр* |
| **Обсяг дисципліни** | *5,5 кредитів ECTS (165 годин). Лекцій 36 годин, Лабораторні роботи (комп'ютерний практикум) - 54 годин, Самостійна робота студента - 75 годин* |
| **Семестровий контроль/ контрольні заходи** | *Екзамен* |
| **Розклад занять** | *http://rozklad.kpi.ua/* |
| **Мова викладання** | *Українська* |
| **Інформація про керівника курсу / викладачів** | *Керівник курсу: завідувач каф. ОТ, д. т. н., проф. Стіренко С. Г., sergii.stirenko@gmail.com*<br><br>*Лектор: старший викладач кафедри обчислювальної техніки Шемсединов Т.Г. timur.shemsedinov@gmail.com*<br>*Лабораторні: асистент кафедри обчислювальної техніки Кухар В.В. kukhar.vitalii@gmail.com* |
| **Розміщення курсу** | https://github.com/HowProgrammingWorks |

### Програма навчальної дисципліни

#### 1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

**Метою** навчальної дисципліни є формування у студентів здатностей (компетентностей) програмування, вільного володіння синтаксисом щонайменше однієї мови програмування, розуміння базових парадигм та концепцій. За результатами вивчення дисципліни студент має бути здатними вирішувати професійні завдання та володіти такими компетентностями:

- Здатність до абстрактного мислення, аналізу та синтезу(ЗК01)
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК06)
- Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення (ФК01)

- Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування (ФК02)
- Здатність розробляти архітектури, модулі та компоненти програмних систем (ФК03)
- Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних (ФК07)
- Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення (ФК08)
- Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя (ФК10)
- Здатність до алгоритмічного та логічного мислення (ФК14)

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі програмні результати навчання:

- Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки (ПРН01)
- Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення (ПРН03)
- Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення (ПРН07)
- Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань (ПРН13)
- Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних (ПРН18)

## 2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Дисципліни, до яких готує цей курс: Основи програмування. Частина 2, Компоненти програмної інженерії. Частини 1 та 2, Основи розробки програмного забезпечення на платформі Node.Js, Системне програмування, Моделювання програмного забезпечення.

## 3. Зміст навчальної дисципліни

Тема 1. Моделювання: абстракції та повторне використання
Тема 2. Алгоритм, программа, синтаксис, мова
Тема 3. Декомпозиція та поділ відповідальності
Тема 4. Порівняння мов та парадигм програмування
Тема 5. Налагодження середовища розробки: Node.js, npm, git, eslint
Тема 6. Консоль та командний рядок у JavaScript и Node.js
Тема 7. Значення, ідентифікатор, змінна та константа, літерал, присвоєння
Тема 8. Типи даних, скалярні, посилання та структурні типи

Тема 9. Робота із строками, шаблонами та юнікодом у JavaScript
Тема 10. Оператор та вираз, блок коду, функція, цикл, умова
Тема 11. Контекст та лексичне оточення
Тема 12. Цикли та ітерування
Тема 13. Процедурна парадигма, виклик, стек та куча
Тема 14. Рекурсія, ітератори та генератори
Тема 15. Структури даних: Масив, список, множина, кортеж
Тема 16. Словник, хеш-таблиця та асоціативний масив
Тема 17. Функція вищого порядку, чиста функція, побічні ефекти
Тема 18. Замикання, функції зворотного виклику, обгортки та події
Тема 19. Функції зворотнього виклику (callback) та різні їх контракти
Тема 20. Події, таймери та EventEmitter
Тема 21. Часткове застосування та каррування, композиція функцій
Тема 22. Чейнінг для методів та функцій
Тема 23. Домішки (mixins)
Тема 24. Мемоізація функцій та кешування результатів
Тема 25. Exceptions та обробка помилок
Тема 26. Proxy и Symbol в JavaScript
Тема 27. Серіалізація та десеріалізація даних
Тема 28. Типізовані масиви, атомарні операції

## 4. Навчальні матеріали та ресурси

***Базова:***

1. Шемсединов Т.Г., Нечай Д.О., Кухар В.В., Орленко О.А., Голіков О.Г., Білочуб М.М., Духін В., Іванова Л.А., Чорненький А.Ю. та інші. Приклади коду та приклади проектів [Електронний ресурс] розміщено за адресою: https://github.com/HowProgrammingWorks/
2. Refactoring: Improving theDesign of Existing Code // MartinFowler
3. CleanCode: A Handbook of Agile Software Craftsmanship // Robert C. Martin
4. Introduction to Algorithms, 3rd Edition // Thomas H. Cormen

***Додаткова:***

1. Шемсединов Т.Г., Нечай Д.О., Кухар В.В., Орленко О.А., Голіков О.Г., Білочуб М.М., Духін В., Іванова Л.А., Чорненький А.Ю. та інші. Технологічний стек Metarhia[Електронний ресурс] розміщено за адресою: https://github.com/metarhia/
2. Algorithms Unlocked // Thomas H. Cormen
3. The Art of Computer Programming // Donald Knuth

## Навчальний контент

### 5. Методика опанування навчальної дисципліни (освітнього компонента)

Основні завдання циклу лабораторних занять (комп'ютерного практикуму) -надбання практичних навичок використання базових концепцій програмування, як то, ідентифікатори, типи та колекції, цикли та функції, та їх реалізації за допомогою мов JavaScript, Python, C++ (або інша на вибір).

| № з/п | Назва лабораторної роботи (комп'ютерного практикуму) | Кількість ауд. годин |
|---|---|---|
| 1 | Змінні та типи даних | 3 |
| 2 | Базовий синтаксис JavaScript | 4 |
| 3 | Функції та методи | 8 |
| 4 | Цикли, ітерування | 8 |
| 5 | Замикання та чеїнінг | 7 |
| 6 | Композиція функцій | 8 |
| 7 | Робота з масивами | 8 |
| 8 | Функції вищого порядку | 8 |
| | **Разом:** | 54 |

### 6. Самостійна робота студента

У процесі виконання індивідуальних завдань студенти повинні опрацьовувати знання, отримані під час лекцій та самостійної роботи, самостійно вивчати визначені теми, поглиблювати свої знання для подальшого навчання. Самостійна робота студентів полягає в наступному:

- підготовці до лекційних занять по вивченню попереднього лекційного матеріалу;
- розглядання та модифікація прикладів коду та проектів з наданих викладачем gitрепозиторіїв;
- виконання лабораторних робіт з вивченням теорії та реалізацією наведеної теми у програмному коді;
- підготовка та участь у обговоренні тем на семінарах;
- пір-рев'ю роду колег студентів;
- підготовка та захищення коду на код-рев'ю викладача.

| № з/п | Назва теми, що виноситься на самостійне опрацювання | Кількість годин СРС |
|---|---|---|
| 1 | Налаштування середовища розробки | 12 |
| 2 | Робота з документацією та специфікацією мови програмування ECMA Script (JavaScript) | 12 |
| 3 | Перевірка коду за допомогою лінтерів (eslint та prettier) | 12 |

## Політика та контроль

### 7. Політика навчальної дисципліни (освітнього компонента)

Під час занять з навчальної дисципліни студенти повинні дотримуватись певних дисциплінарних правил:

1) так використовувати чат-групи, репозиторії та середовище виконання, щоб не створювати проблем з зайвими нотифікаціями іншим учасникам учбового процесу та викладачеві;

2) вчасно виконувати роботу та коммітити все кожного дня у систему контролю версій, не створювати багато коммітів у одному пул-реквесту, не створювати великих коммітів, розділяти все на окремі тематичні комміти, чітко та зрозуміло описувати комміти та пул-реквести, додавати теги для лінкуванняissues, PR, та коммітів у репозиторіях;

3) не допускаються використання піратських копій середовищ розробки, операційних систем чи інших засобів розробки та розгортання як на своїх обчислювальних засобах, так і на хмарних;

4)спамити, постити забагато мемів та стікерів у групах та репозиторіях;

5) якщо виникає питання, потрібно спочатку шукати у матеріалах курсу, потім у інтернеті, потім запитати у асистентів та інших студентів, а потім вже, тільки коли рішення не знайдено чи воно незрозуміле, турбувати викладача;

Лабораторні роботи здаються тільки через Github у відкритому коді (до коду студенти додають ліцензію MIT), а репозиторій повинен мати історію розробки, щоб викладач міг простежити послідовність написання коду та авторство. Розробка відбуваєтья у feature-гілках git, після чого перевірка коду проходить через PR та включає рев'ю, яке викладач робить у пул-реквесті. Історія рев'ю залишається у Github аккаунті студента.

### 8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Види контролю з навчальної дисципліни «Основи програмування. Частина 1» включають:

***Лабораторні роботи***

Заплановано самостійне виконання 6 лабораторних робіт.

Теми лабораторних робіт узгоджені у часі та за змістом з темами лекцій

***Поточний контроль***:

Передбачено 2 попередні рев'ю кода за курс, які повністю охоплюють розглянуту на лекціях тематику навчальної дисципліни. До рев'ю коду викладач може додавати теоретичні питання, які розкривають розуміння теми студентом.

***Семестровий контроль***

Остаточний рев'ю коду робиться за декілька підходів, але по дедлайну пул-реквест зачиняється, та фіксуються усі зауваження, що виправлені чи не виправлені.

***Екзамен***

проводиться у вигляді співбесіди зі студентом для об'єктивного визначення рівня знань, умінь та практичних навичок, отриманих за семестр чи парного лайвкодінгу за викладачем чи асистентом, парного кодінгу з іншим студентом чи повернення до написаного за семестр коду та усунення його недоліків чи обговорення його особливостей.

Рейтинг студента з кредитного модуля складається з балів, які він отримує за види робіт відповідно до таблиці 4.

Таблиця 4

Оцінювання окремих видів навчальної роботи студента

| Розділ 1,2 | | Розділ 3,4 | |
|---|---|---|---|
| Вид навчальної роботи | Максимальна кількість балів | Вид навчальної роботи | Максимальна кількість балів |
| Виконання та захист лабораторної роботи № 1 | 5 | Виконання та захист лабораторної роботи № 4 | 5 |
| Виконання та захист лабораторної роботи № 2 | 5 | Виконання та захист лабораторної роботи № 5 | 5 |
| Виконання та захист лабораторної роботи № 3 | 5 | Виконання та захист лабораторної роботи № 6 | 5 |
| Поточний код-рев'ю №1 | 5 | Поточний код-рев'ю №2 | 5 |
| | | Фінальний код-рев'ю | 10 |
| | | Екзамен | 50 |
| | | **Усього за семестр** | 100 |

Разом за лабораторні роботи (максимальна кількість балів) – 30

Індивідуальний семестровий рейтинг студента (підсумкова семестрова рейтингова оцінка **RD**) є сумою балів, отриманих студентом протягом семестру за участі у семінарах, рев'ю коду, обговореннях.

Усі студенти, незалежно від того, чи вони виконали всі умови допуску до семестрової атестації з кредитного модуля та мають рейтингову оцінку не менше 60 балів, проходять співбесіду з викладачем.

Необхідною умовою допуску студента до екзамену є його індивідуальний семестровий рейтинг (**RD**) не менший, ніж 30% від максимуму балів, тобто 30 балів, здані 4 лабораторні роботи та наявність однієї позитивної атестації в семестрі. При невиконанні хоча б однієї зі згаданих умов студент до екзамену не допускається.

Сума підсумкової семестрової (**RD**) та екзаменаційної рейтингових оцінок у балах становить підсумкову семестрову рейтингову оцінку, яка перераховується в оцінки за національною шкалою та шкалою ECTS (табл. 5).

Таблиця 5

Відповідність рейтингових балів оцінкам за університетською шкалою

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

**Робочу програму навчальної дисципліни (силабус):**

**Складено** старший викладач кафедри обчислювальної техніки Шемсединов Т.Г.

**Ухвалено** кафедрою обчислювальної техніки (протокол № 10 від 25.05.2022)

**Погоджено** Методичною комісією факультету (протокол № 10 від 09.06.2022)