**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ НАУК, ФІЗИКИ ТА МАТЕМАТИКИ**

**КАФЕДРА ІНФОРМАТИКИ, ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ**

|  |  |
| --- | --- |
|  | ЗАТВЕРДЖЕНОна засіданні кафедри ….…протокол від …. …. 2020 р. № … завідувач кафедри\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (проф. В. Песчаненко) |

**СИЛАБУС ОСВІТНЬОЇ КОМПОНЕНТИ**

**Теорія програмування**

Освітня програма Комп’ютерні науки першого (бакалаврського) рівня

Спеціальність 122 Комп’ютерні науки

Галузь знань 12 Інформаційні технології

Херсон 2020

1. **Опис курсу**

|  |  |
| --- | --- |
| **Назва освітньої компоненти** | Теорія програмування |
| **Тип курсу** | Фундаментальна компонента |
| **Рівень вищої освіти** | Перший (бакалаврський) рівень освіти |
| **Кількість кредитів/годин** | 3 кредити / 90 годин |
| **Семестр** | 8 семестр |
| **Викладач** | Михайло Львов (Mihael Lvov), професор кафедри<https://orcid.org/0000-0002-0876-9928>Юлія Тарасіч (Yuliia Tarasich), викладач кафедри<https://orcid.org/0000-0002-6201-4569>  |
| **Посилання на сайт** | http://ksuonline.kspu.edu/course/view.php?id= |
| **Контактний телефон, мессенджер** | (0552)326768<https://t.me/kipiek> |
| **Email викладача:** | lvov@ksu.ks.ua  yutarasich@gmail.com  |
| **Графік консультацій** | Четвер, 15:00-16:00, ауд. 513а (519) або за призначеним часом |
| **Методи викладання** | лекційні заняття, лабораторні роботи, презентації, тестові завдання, індивідуальні завдання |
| **Форма контролю** | Екзамен |

1. **Анотація дисципліни:** курс “Теорія програмування” є складовою частиною циклу дисциплін, що забезпечує підготовку фахівців з теоретичного програмування при розробці ефективних комп’ютерних програм. Програма передбачає комплексне вивчення побудування та перетворення математичних моделей комп’ютерних програм, оптимізацію програм, аналіз та синтез програм, верифікацію програм.
2. **Мета та завдання дисципліни:**

Мета дисципліни: Формування наукової теоретичної точки зору на комп’ютерні програми як на об’єкт досліджень, формування представлень про основні задачі та методи досліджень комп’ютер­них програм, визначення ролі та місця теорії програмування в системі наукових знань про комп’ютерні програми, зв’язку між практикою та теорією програмування.

Завдання:

оволодіння знаннями про використання методів теоретичного програмування при розробці ефективних комп’ютерних програм . Формування уявлень про такі основні задачі теорії програмування: побудування та перетворення математичних моделей комп’ютерних програм, оптимізацію програм, аналіз та синтез програм, верифікацію програм.

1. **Програмні компетентності та результати навчання**

**Після успішного завершення дисципліни здобувач формуватиме наступні програмні компетентності та результати навчання:**

**Інтегральна компетентність** - Здатність розв’язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій.

**Загальні компетентності**:

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

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

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК9. Здатність працювати в команді.

**Фахові компетентності**:

ФК1. Здатність до математичного формулювання та дослідження неперервних та дискретних математичних моделей, обґрунтовування вибору методів і підходів для розв’язування теоретичних і прикладних задач у галузі комп’ютерних наук, аналізу та інтерпретування.

ФК8. Здатність проєктувати 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.

ФК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

**Програмні результати навчання:**

ПРН3. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН 5. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об’єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення

ПРН 12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПРН 14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

1. **Структура курсу**

|  |  |  |  |
| --- | --- | --- | --- |
| **Кількість кредитів/годин** | **Лекції (год.)** | **Лабораторні роботи (год.)** | **Самостійна робота (год.)** |
| 3 кредити / 90 годин | 18 | 22 | 50 |

1. **Технічне й програмне забезпечення/обладнання**

Комп’ютерні аудиторії кафедри

Програмне забезпечення:

1. **Політика курсу**

Для успішного складання підсумкового контролю з дисципліни вимагається 100% очне або дистанційне відвідування всіх занять. Пропуск понад 25% занять без поважної причини буде оцінено як FX.

Високо цінується академічна доброчесність. До всіх студентів освітньої програми відбувається абсолютно рівне ставлення. Навіть окремий випадок порушення академічної доброчесності є серйозним проступком, який може призвести до несправедливого перерозподілу оцінок і, як наслідок, загального рейтингу студентів. Мінімальне покарання для студентів, яких спіймали на обмані чи плагіаті під час тесту чи підсумкового контролю, буде нульовим для цього завдання з послідовним зниженням підсумкової оцінки дисципліни принаймні на одну літеру. Будь ласка, поставтесь до цього питання серйозно та відповідально.

Для поточного контролю знань студентів з навчальної дисципліни використовуються такі методи:

* на лекційних заняттях проводиться контроль присутності студентів та контроль якості конспектів лекцій;
* на лабораторних заняттях проводиться контроль готовності до заняття шляхом тестового експрес-опитування, а також шляхом захисту звітів з лабораторної роботи у вигляді співбесіди;
* контроль самостійної роботи проводиться у вигляді співбесіди на задану тему;
* оцінка модульних контрольних робіт (тестування);
* підсумковий контроль проводиться в кінці семестру у вигляді заліку.

Усі форми контролю включено до 100-бальної шкали оцінювання. Оцінювання результатів поточної роботи (завдань, що виконуються на лабораторних заняттях, результати самостійної роботи студентів) проводиться за такими критеріями:

Лабораторні роботи (у % від кількості балів, виділених на завдання із заокругленням до цілого числа):

0 % – завдання не виконано;

40% – завдання виконано частково та містить суттєві помилки методичного або розрахункового характеру;

60% – завдання виконано повністю, але містить суттєві помилки у розрахунках або в методиці;

80% – завдання виконано повністю і вчасно, проте містить окремі несуттєві недоліки (розмірності, висновки, оформлення тощо);

100% – завдання виконано правильно, вчасно і без зауважень.

1. **Схема курсу**

## Модуль 1. Моделі програм та задача економії пам’яті

**.**

**Тема 1. Предмет теорії програмування. Програми та їх математичні моделі. (л. - 2 год., лаб. – 2 год.)**

* Програми та їх графічні моделі. Графічна модель програми: U-Y схема програми.
* Програми та їх алгебраїчні моделі. Алгоритмічна алгебра Глушкова.
* Моделі пам’яті та моделі даних .
* Моделі обчислень та управління.
* Семантика програм.

## Тема 2. Задача економії пам’яті в операторних схемах. (л. - 2 год., лаб. – 4 год.)

* Аналіз лінійних програм та задача економії пам’яті.
* Приклади розв’язання в задачі економії пам’яті в лінійних програмах.
* Операторна схема лінійної програми як її математична модель. Інформаційні зв’язки та перетини. Зв’язки. Ширина перетинів.
* Алгоритм економії пам’яті у лінійних програмах
* Програми загального вигляду та їх операторні схеми.
* Алгоритм економного розподілу пам’яті у неформальному викладенні. Зведення задачі економного розподілу пам’яті до задачі розфарбування графа.

## Тема 3. Операторні схеми програм як їх математична модель. (л. - 2 год., лаб. – 4 год.)

* Побудування U-Y схем програм та алгебраїчних моделей програм.
* Взаємні перетворення графічних та алгебраїчних моделей програм.

## Тема 4. Алгоритм економного розподілу пам’яті. (л. - 2 год., лаб. – 2 год.)

* Етапи трансляції символічної програми в машинну.
* Інформаційні зв’язки та січення.
* Детальна процедура економії.
* Розширення інформаційних зв’язків .

**Модуль 2. Схеми Янова та їх перетворення. : Аналіз та верифікація програм.**

**Тема 1. Логічні оператори в програмах. Постановка задачі перетворення схем Янова (л. - 2 год., лаб. – 2 год.)**

* Логічні оператори в програмах. Уточнення логічних умов в операторах мови програмування.
* Перетворення логічних операторів за правилами математичної логіки.
* Попередня постановка задачі перетворення схем Янова.
* Означення схем Янова. Приклад

**Тема 2. Схеми Янова та їх еквівалентності (л. - 2 год., лаб. – 2 год.).**

* Функціональна та операційна еквівалентність схем Янова. Теорема про тотожність понять функціональної та операційної еквівалентності
* Формальна еквівалентність схем Янова. Теорема про тотожність понять формальної та операційної еквівалентності.

## Тема 3 Числення рівносильних перетворень схем Янова (л. - 2 год., лаб. – 2 год.).

* Поняття досягненості та продуктивності у схемі Янова.
* Алгоритм верхньої розмітки схеми Янова.
* Алгоритм нижньої розмітки схеми Янова.
* Аксіоми та правила виводу числення схем Янова.
* Теореми про коректність  числення схем Янова.

## Тема 4. Задачі автоматичної генерації інваріантів програм та аналізу програм. Розв’язання задачі генерації інваріантних рівностей у поліноміально-визначених програмах. (л. - 4 год., лаб. – 4 год.).

* Побудування U-Y схем програм та алгебраїчних моделей програм.
* Доведення правильності програм методом Хоара.

Доведення інваріантності рівностей та побудування інваріантних рівностей у поліноміально-визначених програмах.

1. **Система оцінювання та вимоги: форма (метод) контрольного заходу та вимоги до оцінювання програмних результатів навчання**

**Модуль 1. Моделі програм та задача економії пам’яті.**

Форма (метод) контрольного заходу, критерії оцінювання та бали

Лабораторні роботи – 30 балів (по 5 балів за 6 лабораторних робіт)

Тест за модуль - до 5 балів

**Модуль 2. Схеми Янова та їх перетворення. Аналіз та верифікація програм.**

Форма (метод) контрольного заходу, критерії оцінювання та бали

Лабораторні роботи – 25 балів (по 5 балів за 5 лабораторних робіт)

Підсумковий (екзаменаційний) тест за дисципліну – 40 балів

Студенти можуть отримати до 10% бонусних балів за виконання індивідуальних завдань, участь у конкурсах наукових робіт, предметних олімпіадах, конкурсах, неформальній та інформальній освіті (зокрема, COURSERA та ін.).

***У випадку переходу університету на дистанційну форму навчання лабораторні роботи виконуються згідно розкладу занять та надсилаються кожним студентом групи у виді архіву(файлу) на наступний день після лабораторної роботи на електронну пошту викладача, що проводить практичні заняття. Викладач протягом тижня перевіряє надіслані лабораторні роботи та виставляє оцінки в електронний журнал.***

1. **Список рекомендованих джерел (наскрізна нумерація)**
2. Ершов А.П. Введение в теоретическое программирование. (беседы о методе). М.:1977 г., 2888 с. ил.
3. Грис Д. Наука програмирования:Пеp.с англ.-М.:Миp,1984.-416 с.:ил.
4. Дейкстра.Э. Дисциплина программирования: пер. с англ. М.:Мир, 1976).
5. Дал У.,Дейкстра Э., Хоор К. Структурное програмирование: пеp. с англ.-М.:Миp,1975.-248 с.:ил.-(Математическое обеспечение ЭВМ). Шоу А. Логическое проектирование операционных систем. – М.: Мир, 1981. – 360 с.
6. Годлевский А.Б., Капитонова Ю.В., Кривой С.Л., Летичевский А.А. Итеративные методы анализа программ // Кибернетика. – 1984. – №2, С.23-28.
7. A.Letichevsky, M.Lvov. Discovery of invariant Equalities in Programs over Data fields. Applicable Algebra in En­­­gi­neering, Communi­ca­tion and Computing. – 1993. – №4. – pp. 21-29
8. Львов М.С. Инвариантные равенства малых степеней в программах, опеределенных над полем. // Кибернетика. – 1988. - №1. – С. 108 – 110.
9. Львов С.М. О реализации вычислений в задачах анализа программ, определенных над векторными пространствами // Проблемы программирования – 2004.-№№2-3. Специальный выпуск. С. 95-101.
10. Львов М.С. Инвариантные неравенства в програм­мах, интерпретированных над упоря­до­ченными полями. // Кибернетика. – 1986. – №5. – С.22-27
11. Müller-Olm M., Seidl H. Computing polynomial program invariants. // Inf. Process. Lett. September 2004.- Vol 91, Issue 5. Elsevier North-Holland, Inc.  Amsterdam. - pp. 233-244.
12. Sankaranarayanan S., Sipma H., Manna Z. Non-linear loop invariant generation using Gröbner bases.// Proc. of Symposium on Principles of Programming Languages. - Venice, Italy, January 14-16, 2004. ACM: New York, NY, USA, 2004.- pp. 318-329.
13. Rodríguez-Carbonell E., Kapur D. Automatic generation of polynomial loop invariants: algebraic foundations. // Proc. Of International Symposium on Symbolic and Algebraic Computation.- Santander, Spain, July 4-7, 2004.- ACM: New York, NY, USA 2004.- pp. 266-273.
14. Rodríguez-Carbonell E., Kapur D. Automatic generation of polynomial invariants of bounded degree using abstract interpretation. // Sci. Comput. Program. Volume 64 ,  Issue 1  (January 2007).- Elsevier North-Holland, Inc.  Amsterdam, The Netherlands.- 2007.-pp.54-75.