КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«____»_____2019 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ТИПИ В МОВАХ ПРОГРАМУВАННЯ
для студентів

галузь знань	**12 Інформаційні технології**
спеціальність	**121 Інженерія програмного забезпечення**
освітній рівень	**магістр**
освітня програма	**Програмне забезпечення систем**
вид дисципліни	**обов'язкова**

Форма навчання	**денна**
Навчальний рік	**2019/2020**
Семестр	**2**
Кількість кредитів ECTS	**3**
Мова викладання, навчання та оцінювання	**українська**
Форма заключного контролю	**залік**

Викладач: **к. ф.-м. н., доцент Ченцов О.І.** (лекції).

Пролонговано: на 20__/20__ н. р. _____(_____) «__»___ 20__ р.

на 20__/20__ н. р. _____(_____) «__»___ 20__ р.

КИЇВ – 2019

Розробник: Ченцов Олексій Ілліч, к. ф.-м. н., доцент, доцент кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол № __ від «___» _____2019 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «____» _____ 2019 року №___

Голова науково-методичної комісії _____ Л.Л. Омельчук

«_____» _____ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «____» _____ 2019 року №___

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета дисципліни – вивчення принципів, що закладаються в сучасні мови програмування для забезпечення розробки надійніших та якісніших програмних систем.

2. Попередні вимоги до опанування навчальної дисципліни. Для успішного вивчення дисципліни «Типи в мовах програмування» студенти повинні відповідати наступним вимогам:

1. **Знати:** елементарні поняття з дискретної математики, основи об'єктно-орієнтованого програмування.
2. **Вміти:** програмувати на імперативній мові програмування.
3. **Володіти елементарними навичками:** з розробки програмного забезпечення.

3. Анотація навчальної дисципліни. Навчальна дисципліна «Типи в мовах програмування» є складовою освітньо-наукової програми підготовки фахівців за другим (магістерським) рівнем вищої освіти у галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення в рамках освітньо-наукової програми «Програмне забезпечення систем».

Дана дисципліна є нормативною навчальною дисципліною в рамках освітньої програми «Програмне забезпечення систем». Викладається у 2 семестрі в **обсязі – 90 год. (3 кредити ECTS)** зокрема: лекції – 28 год., консультації – 2 год., самостійна робота – 60 год. У курсі передбачено 3 змістових частини, 3 контрольні роботи, та одне тестування. Завершується дисципліна – **заліком.**

В результаті вивчення навчальної дисципліни студент повинен:

знати: теоретичні основи систем типів сучасних мов програмування, підходи до обґрунтування формальних властивостей обчислювальних систем.

вміти: ефективно користуватися параметризованими програмними компонентами на рівні прикладного програміста, застосовувати нетривіальні прийоми програмування, що базуються на використанні типів.

Дисципліна «Типи в мовах програмування» є базовою для вивчення дисципліни «Актуальні проблеми об'єктно-орієнтованого програмування» другого (магістерського) рівня вищої освіти у галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення, в рамках освітньо-наукової програми «Програмне забезпечення систем».

4. Завдання (навчальні цілі). Основними завданнями дисципліни «Типи в мовах програмування» є набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень в області програмування відповідно до освітньої кваліфікації магістр з інженерії програмного забезпечення. Зокрема, розвивати:

- Здатність проведення теоретичних та прикладних досліджень на відповідному рівні (ЗК-3).
- Здатність проектувати програмне забезпечення, включаючи проведення моделювання його архітектури, поведінки та процесів функціонування окремих підсистем і модулів (СК-3).
- Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення (СК-4).

5. Результати навчання за дисципліною.

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	**Результат навчання**			
РН1.1	Знати теоретичні основи систем типів сучасних мов програмування	Лекції.	Контрольна робота, тест 60% балів.	25%
РН1.2	Знати аспекти реалізації систем типів	Лекції.	Контрольна робота, тест 60% балів.	25%
РН2.1	Вміти використовувати пов’язані з типами конструкції у сучасному інструментарії програмування.	Самостійна робота.	Захист індивідуального завдання, 60% балів.	30%
РН2.2	Вміти застосовувати прийоми та стилі програмування, що базуються на використанні типів	Лекції.	Контрольна робота, 60% балів.	20%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Результати навчання дисципліни / **Програмні результати навчання**	**РН1.1**	**РН1.2**	**РН2.1**	**РН2.2**
ПРН-7. Обґрунтовано обирати парадигми і мови програмування для вирішення прикладних завдань; застосовувати на практиці системні та спеціалізовані засоби, компонентні технології (платформи) та інтегровані середовища розробки програмного забезпечення.	+		+	+
ПРН-8. Проводити аналітичне дослідження параметрів функціонування програмних систем для їх валідації та верифікації, а також проводити аналіз обраних методів, засобів автоматизованого проектування та реалізації програмного забезпечення.	+	+		

7. Схема формування оцінки.

7.1 Форми оцінювання студентів.

Семестрове оцінювання:

1. Контрольна робота 1: РН1.1 – **10 балів/5 балів**.
2. 2. Контрольна робота 2: РН1.2 – **10 балів/5 балів**.

3. Контрольна робота 3: РН2.2 – **20 балів/10 балів**.
4. Тест: РН1.1, РН1.2 – **30 балів/15 балів**.
5. Індивідуальне завдання – **30 балів/15 балів**.

Підсумкове оцінювання:

- залік проводиться за сумарними результатами контрольних заходів та роботи на заняттях;
- студентам, які набрали впродовж семестру 60 балів і більше, залік **виставляється**;
- студенти, які не набрали впродовж семестру 60 балів, до заліку **не допускаються**.

7.2 Організація оцінювання.

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до 4 тижня семестру.
2. Контрольна робота 2: до 7 тижня семестру.
3. Контрольна робота 3: до 14 тижня семестру.
4. Індивідуальне завдання: до 11 тижня семестру.

Якщо студенти з поважних причин, які підтверджено документально, були відсутні під час написання контрольної роботи, вони мають право на одне перескладання з можливістю отримання максимальної кількості балів. Термін перескладання визначається викладачем.

Студенти мають право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

У разі неякісного виконання індивідуального завдання, викладач має право не зарахувати його, або знизити за нього бали.

Студенти мають право здавати індивідуальне завдання після закінчення визначеного для нього терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну здачі.

7.3 Шкала відповідності оцінок.

Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій.

№ лекції	Назва теми/лекції	Кількість годин	
		Лекції	Самостійна робота
Частина 1. Базова теорія та реалізація простої системи типів.			
1	**Тема 1.** Вступ до дисципліни.	2	2
2	**Тема 2.** Мовний рівень обчислювальної системи.	2	2
3	**Тема 3**. Семантика простої обчислювальної системи.	2	2
4	**Тема 4**. Проста система типів, її безпека.	2	2
Контрольна робота 1			2
5	**Тема 5**. Нетипізоване лямбда-числення.	2	4

6	**Тема 6.** Лямбда-числення з простою типізацією.	2	2
Контрольна робота 2			2
Всього по частині 1		12	18
Частина 2. Поліморфізм.			
7	**Тема 7.** Різновиди поліморфізму.	2	3
8	**Тема 8**. Проблема реконструкції типів.	2	2
9	**Тема 9**. Поліморфізм Гіндлі-Мілнера.	2	4
10	**Тема 10**. Поліморфізм вищих рангів.	2	5
Індивідуальне завдання			8
Всього по частині 2		8	22
Частина 3. Параметризоване програмування.			
11	**Тема 11.** Параметризоване програмування на основі концептів.	2	4
12	**Тема 12.** Дженерики та їх використання.	2	4
13	**Тема 13.** Програмування параметризоване за класами алгебр.	2	5
14	**Тема 14.** Функторна теорія індуктивних типів даних.	2	5
Контрольна робота 3, Тест			2
Всього по частині 3		8	20
Консультації		2	
ВСЬОГО		**28**	**60**

Загальний обсяг **– 90** год., в тому числі:
Лекції **– 28** год.
Консультації – **2** год.
Самостійна робота – **60** год.

Теми, винесені на самостійне вивчення.

1. Поліморфізм сімейств.
2. Обмеження на типи у стандартній бібліотеці C++.
3. Використання generics при програмуванні на мові Java.
4. Проблема бінарного методу.

Умови індивідуального завдання.

Методичні матеріали з дисципліни, варіанти індивідуального завдань доступні за посиланням: https://drive.google.com/open?id=0B-BUpwNPP_9JX1o0MFZIbVJDWlU

9. Рекомендовані джерела:

Основні:

1. Pierce B. Types and Programming Languages / Pierce B. – Massachusetts: MIT Press, 2002. – 645p.

2. Cardelli L. On Understanding Types, Data Abstraction, and Polymorphism / Luca Cardelli and Peter Wegner // ACM Comput. Surv. – 1985. – vol. 17, no.4. — Pp. 471–522.
3. Stepanov A. Elements of programming / Alexander Stepanov, Paul Mc Jones. – Addison-Wesley, 2009. – 288 p.
4. Stepanov A. From mathematics to generic programming / Alexander A. Stepanov, Daniel E. Rose. – Addison-Wesley, 2014. – 320 p.
5. Backhouse R. et al. Generic Programming: An Introduction // Advanced Functional Programming Third International School. Braga, Portugal. – 1998. – Pp. 28–115.
6. Gibbons J. Datatype-generic programming / J. Gibbons. // Lecture Notes in Computer Science. – 2007. – vol. 4719. – Pp. 1–71.

Додаткові:

1. Siek J. A language for generic programming: Ph.D. thesis / J. Siek; Indiana University. – 2005. – 260 p
2. Generic Programming / [Mehdi Jazayeri, R. Loos, David Musser, and Alexander Stepanov] // Report of the Dagstuhl Seminar on Generic Programming. – Dagstuhl Schloss, Germany, 1998.
3. On binary methods / [Kim Bruce, Luca Cardelli, et al.] // Theory and practice of object systems. – 1995. – Vol. 1, no.3. – Pp. 221-242.
4. Ernst E. Family polymorphism / E. Ernst // Lecture Notes in Computer Science. – 2001. – Vol. 2072. – Pp. 303-326.
5. Madsen O.L. Virtual classes: a powerful mechanism in object-oriented programming / O.L. Madsen, B. Moller-Pedersen // Conference proceedings on Object-oriented programming systems, languages and applications (OOPSLA 89). – New Orleans, US, 1989. – Pp. 397-406.
6. Strachey C. Fundamental concepts in programming languages / C. Strachey // Lecture Notes, International Summer School in Computer Programming. – Copenhagen, August 1967. – Pp. 1-49.
7. Siek J. Concept Checking: Binding Parametric Polymorphism in C++ / Jeremy Siek, Andrew Lumsdaine // First Workshop on C++ Template Programming. – Erfurt, Germany, 2000.
8. Kapur D. Tecton: a framework for specifying and verifying generic system components: Technical Report / D. Kapur, D. Musser / Department of Computer Science, Rensselaer Polytechnic Institute. – RPI–92–20. – New York, July 1992.
9. Musser D. Generic programming and formal methods / D. Musser // Workshop on The Verification Grand Challenge. – Menlo Park, CA, 2005.
10. An extended comparative study of language support for generic programming / R. Garcia, J. Jaarvi, A. Lumsdaine et al. // Journal of Functional Programming. – 2005. – Pp. 115–134.
11. Sozeau M. First-class type classes / M. Sozeau, N. Oury // Proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics. – 2008. – Pp. 278-293.
12. Schwarzweller Ñ. Towards Formal Support for Generic Programming: Habilitation thesis / Wilhelm-Schickard-Institute for Computer Science, University of Tubingen. – 2003. – 143 pp.

13. Stroustrup B. A Concept Design for the STL. / B. Stroustrup and A. Sutton // Technical Report N3351=12-0041, ISO/IEC JTC1/SC22/WG21. – The C++ Standards Committee, Jan. 2012. – 133 p.

10. Додаткові ресурси.

Приклади контрольних завдань, тестових запитань, перелік тем, що виносяться на перевірку, розміщено за посиланням:

–https://drive.google.com/open?id=1CwoEA2hMxfnxlxgQVnsqzmzt8peQN7Xm