**СИЛЛАБУС**

**Осенний семестр 2023-2024 учебного года**

**Образовательная программа «Интеллектуальные системы управления»**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID и наименование дисциплины** | **Самостоятельная работа обучающегося**  **(СРС)** | | **Кол-во кредитов** | | | **Общее**  **кол-во кредитов** | **Самостоятельная работа обучающегося**  **под руководством преподавателя (СРСП)** |
| **Лекции (Л)** | **Практ. занятия (ПЗ)** | **Лаб. занятия (ЛЗ)** |
| 199376 | 2 | | 30 | 30 | 30 | 9 | 7 |
| **АКАДЕМИЧЕСКАЯ ИНФОРМАЦИЯ О ДИСЦИПЛИНЕ** | | | | | | | |
| **Формат обучения** | **Цикл,**  **компонент** | **Типы лекций** | | **Типы практических занятий** | | **Форма и платформа**  **итогового контроля** | |
| *Оффлайн* | Обязательные дисциплины | Аналитические, проблемные | | Программирование на Python | | Устный экзамен | |
| **Лектор - (ы)** | Кудайкулов Азиз Анарбаевич | | | | |
| **e-mail:** | aziz.kudaikulov@gmail.com | | | | |
| **Телефон:** | 87019640896 | | | | |
| **Ассистент- (ы)** | Кудайкулов Азиз Анарбаевич | | | | |
| **e-mail:** | aziz.kudaikulov@gmail.com | | | | |
| **Телефон:** | 87019640896 | | | | |
| **АКАДЕМИЧЕСКАЯ ПРЕЗЕНТАЦИЯ ДИСЦИПЛИНЫ** | | | | | | | |
| **Цель дисциплины** | **Ожидаемые результаты обучения (РО)\*** | | | | | **Индикаторы достижения РО (ИД)** | |
| Цель курса в овладении основных идей методов, особенностей областей применения и методики использования их как готового инструмента практической работы при проектировании и разработке систем МО, математической обработке данных экономических и других задач, построении алгоритмов и организации вычислительных процессов на ПК. | 1. Уметь реализовывать алгоритмы решения оптимизационных задач, анализировать сложность алгоритма. | | | | | 1.1 Умеет программировать на языке Python. Умеет работать со сторонними библиотеками, такими как: numpy, matplotlib, scipy. | |
| 1.2 Умеет реализовывать алгоритмы решения оптимизационных задач, анализировать сложность алгоритма. | |
| 2. Уметь анализировать задачи оптимизации геометрически. | | | | | 2.1 Умеет визуализировать постановку задач оптимизации. | |
| 2.2 Умеет показать корректность поставленной задачи. | |
| 3. Знать основные алгоритмы решения задач оптимизации. | | | | | 3.1 Знает недостатки и преимущества алгоритмов решения задач, такие как: сложность алгоритма, погрешность вычисления, скорость сходимости, устойчивость. | |
| 3.2 Умеет выбирать или комбинировать алгоритмы для решения конкретных задач. | |
| 4. Уметь применять алгоритмы решения задач оптимизации в случае не определенности некоторых входных параметров задачи. | | | | | 4.1 Умеет оценивать решения задач оптимизации в случае не определенности некоторых входных параметров задачи. | |
| 4.2 Умеет находить эффективные решения задач оптимизации в случае не определенности некоторых входных параметров задачи. | |
| 5. Уметь декомпозировать задачу на более мелкие подзадачи. | | | | | 5.1 Умеет разбивать задачи на подзадачи. | |
| 5.2 Умеет разработать алгоритмы решения задач, состоящих из подзадач. | |
| **Пререквизиты** | Алгоритмы и структуры данных, Линейная алгебра | | | | | | |
| **Постреквизиты** | Дискретная математика и математическая логика, Линейные системы автоматического регулирования, Управление технологическим процессом производства. | | | | | | |
| **Учебные ресурсы** | **Литература:** основная, дополнительная.  1. Kwon R.H. Introduction to Linear Optimization and Extensions with MatLab, CRC Press, 2022. – 356 p.  2. Avraamidou S., Pistikopoulos E. Multi-level Mixed-Integer Optimization, Parametric Programming Approach, de Gruyter, 2022. – 154 p.  3. Khan S., Bari A., Khan F.S. Linear and Integer Programming, Cambridge Scholars Publishing, 2019. – 232 p.  4. Wolsey L.A. Integer Programming, Wiley, 2021. – 360 p.  5. Bertsimas D., Tsitsiklis J.N. Introduction to Linear Optimization, MIT, 1997. – 606 p.  6. Saoub K.R. A Tour Through Graph Theory, CRC Press, 2018. – 321 p.  7. Laaksonen A. Competitive Programmer’s Handbook, Helsinki, 2018. – 296 p.  **Исследовательская инфраструктура**  1. Компьютерный класс  **Профессиональные научные базы данных**  1. Control theory and technology - https://link.springer.com/journal/11768  2**.** Journal of Optimization Theory and Applications - <https://link.springer.com/journal/10957>  3. Automation and Remote Control - <https://link.springer.com/journal/10513>  4. Autonomous Intelligent Systems - <https://link.springer.com/journal/43684>  5. GSTF Journal of Mathematics, Statistics and Operations Research (JMSOR) - https://link.springer.com/journal/40836/volumes-and-issues  **Интернет-ресурсы**  1. https://github.com/azizka85/Linear-Integer-Programming  2. <https://git-scm.com/>  3. <https://code.visualstudio.com/>  4. https://www.python.org/ | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Академическая политика дисциплины** | | Академическая политика дисциплины определяется [Академической политикой](https://univer.kaznu.kz/Content/instructions/%D0%90%D0%BA%D0%B0%D0%B4%D0%B5%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F%20%D0%BF%D0%BE%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B0.pdf) и [Политикой академической честности КазНУ имени аль-Фараби.](https://univer.kaznu.kz/Content/instructions/%D0%9F%D0%BE%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B0%20%D0%B0%D0%BA%D0%B0%D0%B4%D0%B5%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B9%20%D1%87%D0%B5%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B8.pdf)  Документы доступны на главной странице ИС Univer.  **Интеграция науки и образования.** Научно-исследовательская работа студентов, магистрантов и докторантов – это углубление учебного процесса. Она организуется непосредственно на кафедрах, в лабораториях, научных и проектных подразделениях университета, в студенческих научно-технических объединениях. Самостоятельная работа обучающихся на всех уровнях образования направлена на развитие исследовательских навыков и компетенций на основе получения нового знания с применением современных научно-исследовательских и информационных технологий. Преподаватель исследовательского университета интегрирует результаты научной деятельности в тематику лекций и семинарских (практических) занятий, лабораторных занятий и в задания СРСП, СРС, которые отражаются в силлабусе и отвечают за актуальность тематик учебных занятий изаданий.  **Посещаемость.** Дедлайн каждого задания указан в календаре (графике) реализации содержания дисциплины. Несоблюдение дедлайнов приводит к потере баллов.  **Академическая честность.** Практические/лабораторные занятия, СРС развивают у обучающегося самостоятельность, критическое мышление, креативность. Недопустимы плагиат, подлог, использование шпаргалок, списывание на всех этапах выполнения заданий.  Соблюдение академической честности в период теоретического обучения и на экзаменах помимо основных политик регламентируют [«Правила проведения итогового контроля»](https://univer.kaznu.kz/Content/instructions/%D0%9F%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%B8%D1%82%D0%BE%D0%B3%D0%BE%D0%B2%D0%BE%D0%B3%D0%BE%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F%20%D0%9B%D0%AD%D0%A1%202022-2023%20%D1%83%D1%87%D0%B3%D0%BE%D0%B4%20%D1%80%D1%83%D1%81%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5.pdf), [«Инструкции для проведения итогового контроля осеннего/весеннего семестра текущего учебного года»](https://univer.kaznu.kz/Content/instructions/%D0%98%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D0%B8%D1%8F%20%D0%B4%D0%BB%D1%8F%20%D0%B8%D1%82%D0%BE%D0%B3%D0%BE%D0%B2%D0%BE%D0%B3%D0%BE%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F%20%D0%B2%D0%B5%D1%81%D0%B5%D0%BD%D0%BD%D0%B5%D0%B3%D0%BE%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80%D0%B0%202022-2023.pdf), «Положение о проверке текстовых документов обучающихся на наличие заимствований».  Документы доступны на главной странице ИС Univer.  **Основные принципы инклюзивного образования.** Образовательная среда университета задумана как безопасное место, где всегда присутствуют поддержка и равное отношение со стороны преподавателя ко всем обучающимся и обучающихся друг к другу независимо от гендерной, расовой/ этнической принадлежности, религиозных убеждений, социально-экономического статуса, физического здоровья студента и др. Все люди нуждаются в поддержке и дружбе ровесников и сокурсников. Для всех студентов достижение прогресса скорее в том, что они могут делать, чем в том, что не могут. Разнообразие усиливает все стороны жизни.  Все обучающиеся, особенно с ограниченными возможностями, могут получать консультативную помощь по телефону/ е-mail *aziz.kudaikulov@gmail.com.*  **Интеграция МООC (massive open online course).** В случае интеграции МООC в дисциплину, всем обучающимся необходимо зарегистрироваться на МООC. Сроки прохождения модулей МООC должны неукоснительно соблюдаться в соответствии с графиком изучения дисциплины.  **ВНИМАНИЕ!** Дедлайн каждого задания указан в календаре (графике) реализации содержания дисциплины, а также в МООC. Несоблюдение дедлайнов приводит к потере баллов. | | | | |
| **ИНФОРМАЦИЯ О ПРЕПОДАВАНИИ, ОБУЧЕНИИ И ОЦЕНИВАНИИ** | | | | | | |
| **Балльно-рейтинговая**  **буквенная система оценки учета учебных достижений** | | | | | **Методы оценивания** | |
| **Оценка** | **Цифровой**  **эквивалент**  **баллов** | | **Баллы,**  **% содержание** | **Оценка по традиционной системе** | **Критериальное оценивание** – процесс соотнесения реально достигнутых результатов обучения с ожидаемыми результатами обучения на основе четко выработанных критериев. Основано на формативном и суммативном оценивании.  **Формативное оценивание –** вид оценивания, который проводится в ходе повседневной учебной деятельности. Является текущим показателем успеваемости. Обеспечивает оперативную взаимосвязь между обучающимся и преподавателем. Позволяет определить возможности обучающегося, выявить трудности, помочь в достижении наилучших результатов, своевременно корректировать преподавателю образовательный процесс. Оценивается выполнение заданий, активность работы в аудитории во время лекций, семинаров, практических занятий (дискуссии, викторины, дебаты, круглые столы, лабораторные работы и т. д.). Оцениваются приобретенные знания и компетенции.  **Суммативное оценивание** –вид оценивания, который проводится по завершению изучения раздела в соответствии с программой дисциплины.Проводится 3-4 раза за семестр при выполнении СРС. Это оценивание освоения ожидаемых результатов обучения в соотнесенности с дескрипторами. Позволяет определять и фиксировать уровень освоения дисциплины за определенный период. Оцениваются результаты обучения. | |
| A | 4,0 | | 95-100 | Отлично |
| A- | 3,67 | | 90-94 |
| B+ | 3,33 | | 85-89 | Хорошо |
| B | 3,0 | | 80-84 | **Формативное и суммативное оценивание** | **Баллы % содержание** |
| B- | 2,67 | | 75-79 | Активность на лекциях | 5 |
| C+ | 2,33 | | 70-74 | Работа на практических занятиях | 20 |
| C | 2,0 | | 65-69 | Удовлетворительно | Самостоятельная работа | 25 |
| C- | 1,67 | | 60-64 | Проектная и творческая деятельность | 10 |
| D+ | 1,33 | | 55-59 | Неудовлетворительно | Итоговый контроль (экзамен) | 40 |
| D | 1,0 | | 50-54 | ИТОГО | 100 |
| **Календарь (график) реализации содержания дисциплины. Методы преподавания и обучения.** | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Неделя** | **Название темы** | **Кол-во часов** | **Макс.**  **балл** |
| **МОДУЛЬ 1 Линейное программирование** | | | |
| 1 | **Л 1.** Общие задачи линейного программирования. Стандартная форма линейного программирования. | 2 |  |
| **Семинар 1.** Примеры линейного программирования. | 2 | 2 |
| **ЛЗ 1.** Решение задач линейного программирования на Python. | 2 | 8 |
| 2 | **Л 2.** Геометрия допустимого множества. Геометрия оптимальных решении. Геометрическое описание оптимальности. Экстремальные точки и базисные допустимые решения. Генерация базисных допустимых решений. Теорема разложения (представления). Фундаментальная теорема линейного программирования. | 2 |  |
| **СЗ 2.** Примеры генерации базисных допустимых решений. | 2 | 2 |
| **ЛЗ 2.** Генерация базисных допустимых решений на Python. | 2 | 8 |
| **СРС 1.** Групповой проект. Применение симплекс метода для решения реальных задач. |  | 30 |
| 3 | **Л 3.** Симплекс метод. Смежные базисные допустимые решения. Проверка оптимальности базисного допустимого решения. Переход к улучшенному смежному базисному допустимому решению. | 2 |  |
| **СЗ 3.** Примеры использования симплекс метода. | 2 | 2 |
| **ЛЗ 3.** Разработка алгоритма симплекс метода на Python. | 2 | 8 |
| **СРCП 1.** Консультации по выполнению **СРC 1** |  |  |
| 4 | **Л 4.** Генерация начального базисного допустимого решения (методы двух фаз и Big M). Усовершенствованный симплекс-метод. | 2 |  |
| **СЗ 4.** Примеры использования метода двух фаз и Big M. | 2 | 2 |
| **ЛЗ 4.** Разработка усовершенствованного алгоритма симплекс метода на Python. Сложность алгоритма симплекс метода. | 2 | 8 |
| **СРCП 2.** Консультации по выполнению **СРC 1** |  |  |
| 5 | **Л 5.** Теория двойственности. Формирование двойственной задачи для общих линейных программ. Теория слабой и сильной двойственности. Возможности прямой-двойственной оптимизации. Двойственность и симплекс метод. Двойственный симплекс метод. | 2 |  |
| **СЗ 5.** Примеры использования двойственного симплекс метода. | 2 | 2 |
| **ЛЗ 5.** Разработка алгоритма двойственного симплекс метода на Python. | 2 | 8 |
| **СРCП 3.** Консультации по выполнению **СРC 1** |  |  |
| **МОДУЛЬ 2 Алгоритмы линейного и квадратичного программирования** | | | |
| 6 | **Л 6.** Разложение Данцига — Вулфа | 2 |  |
| **СЗ 6.** Примеры использования разложения Данцига — Вулфа. | 2 | 2 |
| **ЛЗ 6.** Разработка алгоритма разложения Данцига — Вулфа на Python. | 2 | 8 |
| 7 | **Л 7.** Методы внутренней точки. Стратегия прямой-двойственной внутренней точки. Метод прямой-двойственной внутренней точки. Предиктор-корректор для метода прямой-двойственной внутренней точки. | 2 |  |
| **СЗ 7.** Примеры использования метода прямой-двойственной внутренней точки. | 2 | 2 |
| **ЛЗ 7.** Разработка алгоритма метода прямой-двойственной внутренней точки на Python. | 2 | 8 |
| **Рубежный контроль 1** | | | **100** |
| 8 | **Л 8.** Квадратичное программирование. | 2 |  |
| **СЗ 8.** Примеры квадратичного программирования. | 2 | 2 |
| **ЛЗ 8.** Решение задач квадратичного программирования на Python. | 2 | 8 |
| **СРC 2.** Групповой проект. Линейное программирование в случае неопределенности входных параметров. |  | 20 |
| 9 | **Л 9.** Линейное программирование в случае неопределенности входных параметров. Стохастическое программирование. | 2 |  |
| **СЗ 9.** Примеры стохастического программирования. | 2 | 2 |
| **ЛЗ 9.** Решение задач стохастического программирования на Python. | 2 | 8 |
| 10 | **Л 10.** Двухэтапное стохастическое программирование с возвратом. Разложение Данцига-Вулфа для стохастического программирования. Метод L-образного контура (генерация ограничений). Интервал неопределенности. | 2 |  |
| **СЗ 10.** Примеры использования метода L-образного контура. | 2 | 2 |
| **ЛЗ 10.** Разработка алгоритма метода L-образного контура на Python. | 2 | 8 |
| **СРCП 4.** Консультации по выполнению **СРС 2** |  |  |
| **МОДУЛЬ 3 Дискретное программирование** | | | |
| 11 | **Л 11.** Общие задачи дискретного программирования. Хорошо решенные задачи | 2 |  |
| **СЗ 11.** Примеры хорошо решенных задач. | 2 | 2 |
| **ЛЗ 11.** Решение задач дискретного программирования на Python. | 2 | 8 |
| 12 | **Л12.** Соответствия и назначения. Дополняющие пути и оптимальность. Максимальное кардинальное совпадение в двудольном графе. | 2 |  |
| **СЗ 12.** Примеры соответствия и назначения. | 2 | 2 |
| **ЛЗ 12.** Реализация алгоритма для нахождения максимального кардинального совпадения в двудольном графе на Python. | 2 | 8 |
| 13 | **Л 13.** Динамическое программирование. | 2 |  |
| **СЗ 13.** Примеры динамического программирования. | 2 | 2 |
| **ЛЗ 13.** Разработка алгоритмов динамического программирования на Python. | 2 | 8 |
| **СРCП 5.** Консультации по выполнению **СРС 2** |  |  |
| 14 | **Л 14.** Алгоритмы вида “Разделяй и властвуй” | 2 |  |
| **СЗ 14.** Примеры использования алгоритма “Разделяй и властвуй” | 2 | 2 |
| **ЛЗ 14.** Разработка алгоритмов вида “Разделяй и властвуй” на Python. | 2 | 8 |
| 15 | **Л 15.** Алгоритм Дейкстра. | 2 |  |
| **СЗ 15.** Примеры на использование алгоритма Дейкстры. | 2 | 2 |
| **ЛЗ 15.** Разработка алгоритма Дейкстра на Python. | 2 | 8 |
| **Рубежный контроль 2** | | | **100** |
| **Итоговый контроль (экзамен)** | | | **100** |
| **ИТОГО за дисциплину** | | | **100** |

**Декан \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Урмашев Б.А.

**Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Мансурова М.Е.

**Лектор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Кудайкулов А.