

СИЛЛАБУС
Осенний семестр 2025 – 2026 учебного года
Образовательная программа «6В06102 Информационные системы»

ID и наименование дисциплины	Самостоятельная работа обучающегося (СРО)	Кол-во кредитов			Общее кол-во кредитов	Самостоятельная работа обучающегося под руководством преподавателя (СРОП)
		Лекции (Л)	Практ. занятия (ПЗ)	Лаб. занятия (ЛЗ)		
92682 Объектно-ориентированное программирование	4	1,7	-	3,3	5	5
АКАДЕМИЧЕСКАЯ ИНФОРМАЦИЯ О ДИСЦИПЛИНЕ						
Формат обучения	Цикл, модуль компонент	Типы лекций	Типы практических занятий	Форма и платформа итогового контроля		
Офлайн	Б ВК	Обзорные лекции основные понятия java	Построение моделей классификации, кластеризации, регрессии на java	Письменный офлайн		
Лектор - (ы)	Байкувеков Мейржан Берикович.					
e-mail:	baikuvekov_meirzhan3@live.kaznu.kz					
Телефон:	+7 702 6478122					
Ассистент- (ы)	Қажыбек Айгерим Маратқызы					
e-mail:	Kazhybek_Aigerim @live.kaznu.kz					
Телефон:	+7 747 983 3787					
АКАДЕМИЧЕСКАЯ ПРЕЗЕНТАЦИЯ ДИСЦИПЛИНЫ						
Цель дисциплины	Ожидаемые результаты обучения (РО)*			Индикаторы достижения РО (ИД)		
Подготовка студентов к глубокому пониманию принципов объектно-ориентированного программирования (ООП) и применению их в языке программирования Java, развитие навыков создания эффективных и надежных программных решений.	1. Понимание и применение основных принципов объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм, абстракция).			1.1 Студент способен объяснить основные принципы ООП.		
				1.2Студент использует концепции инкапсуляции, наследования и полиморфизма в реальных программных проектах.		
	2. Применение основных структур данных в языке Java (массивы, списки, множества, карты).			2.1Студент использует стандартные коллекции Java (ArrayList, HashMap, HashSet).		
				2.2 Студент оптимизирует использование структур данных для решения практических задач.		
	3. Эффективное использование механизма обработки исключений (Exception Handling) в Java.			3.1Студент применяет конструкции try-catch-finally для обработки исключений в коде.		
				3.2 Студент создает и использует собственные исключения для повышения стабильности программ.		
	4. Разработка и тестирование многозадачных программ в Java с использованием многопоточности.			4.1Студент создает многозадачные приложения с использованием потоков в Java.		
				4.2Студент реализует синхронизацию потоков для безопасного выполнения многозадачных операций.		
	5. Создание и внедрение графических интерфейсов пользователя (GUI) с использованием JavaFX.			5.1 Студент разрабатывает графические интерфейсы с использованием JavaFX.		
				5.2Студент разрабатывает интерактивные приложения с графическим интерфейсом.		
Пререквизиты	1. Основы программирования 2. Алгоритмы и структуры данных 3. Основы объектно-ориентированного программирования (ООП)					

	4. Язык программирования Java (основы)
Постреквизиты	1. Разработка веб-приложений на Java 2. Алгоритмы и структуры данных (углубленный курс) 3. Разработка мобильных приложений на Java
Учебные ресурсы	Литература: основная, дополнительная. 1. Шилдт, Г. — Java: Полное руководство. — 10-е изд. — М.: Вильямс, 2020. 2. Гамма, Э., Хелм, Р., Джонсон, Р., Влиссидес, Дж. — Приемы объектно-ориентированного проектирования: Паттерны. — М.: Вильямс, 2020. 3. Лафоре, Р. — Объектно-ориентированное программирование на языке Java. — М.: Диалектика, 2020. 4. Седжвик, Р. — Алгоритмы на Java. — М.: Бином, 2020. 5. Бенжамин, Р. — Профессиональное программирование на Java. — М.: ДМК Пресс, 2021. Исследовательская инфраструктура 1. 1 Язык программирования Java (версии 8+). 2. Библиотеки для ООП и работы с данными: Java Collections, JavaFX, Hibernate. 3. Среда разработки: IntelliJ IDEA, Eclipse, NetBeans. 4. Системы управления базами данных: MySQL, PostgreSQL, Oracle. 5. GitHub – платформа для совместной разработки и хостинга проектов. Профессиональные научные базы данных 1. IEEE Xplore Сайт: IEEE Xplore 2. ACM Digital Library аҗт: ACM Digital Library . 3. SpringerLink Car: SpringerLink 4. ScienceDirect C a r : Science Direct 5. Web of Science Caйт: Web of Science Интернет-ресурсы (не менее 3-5) 1. Stack Overflow 2. GitHub 3. Kaggle 4. Coursera 5. MDN Web Docs 6. W3Schools Программное обеспечение. 1. Java 8+ 2. IntelliJ IDEA 3. Eclipse

Академическая политика дисциплины	Академическая политика дисциплины определяется <u>Академической политикой и Политикой академической честности КазНУ имени аль-Фараби</u>. 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 СРОП, СРО, которые отражаются в силлабусе и отвечают за актуальность тематик учебных занятий и заданий. Посещаемость. Дедлайн каждого задания указан в календаре (графике) реализации содержания дисциплины. Несоблюдение дедлайнов приводит к потере баллов. Академическая честность. Практические/лабораторные занятия, СРО 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 <u>«Правила проведения итогового контроля»</u>, <u>«Инструкции для проведения итогового контроля осеннего/весеннего семестра текущего учебного года»</u>, <u>«Положение о проверке текстовых документов обучающихся на наличие заимствований»</u>. 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 обучающимся 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/ e-mail либо +7 7026478122 посредством видеосвязи в MS Teams <https://us04web.teams.us/j/73674837609?pwd=AlD37HhIShnBYtJOgU7qqmDQ9oK4na.1>

Интеграция MOOC (massive open online course). В случае интеграции MOOC в дисциплину, всем обучающимся необходимо зарегистрироваться на MOOC. Сроки прохождения модулей MOOC должны неукоснительно соблюдаться в соответствии с графиком изучения дисциплины.

ВНИМАНИЕ! Дедлайн каждого задания указан в календаре (графике) реализации содержания дисциплины, а также в MOOC. 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 4 раза за семестр при выполнении СРО. 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			
B-	2,67	75-79			
C+	2,33	70-74			
C	2,0	65-69	Удовлетворительно	Формативное и суммативное оценивание	
C-	1,67	60-64		Баллы % содержание	
D+	1,33	55-59		Активность на лекциях	
D	1,0	50-54		0	
FX	0,5	25-49	Неудовлетворительно	Работа на практических занятиях	
F	0	0-24		20	
				Самостоятельная работа	
				25	
				Проектная и творческая деятельность	
				10	
				Итоговый контроль (экзамен)	
				40	
				ИТОГО	
				100	

Календарь (график) реализации содержания дисциплины. Методы преподавания и обучения.

Неделя	Название темы	Кол-во часов	Макс. балл
МОДУЛЬ 1 Основы ООП и синтаксис языка Java.			
1	Л 1. Принципы ООП (классы, объекты, инкапсуляция, наследование, полиморфизм).	1	
	ЛЗ 1. Написание программы "Hello World" на языке Java.	2	
2	Л 2. Определение классов и создание объектов.	1	
	ЛЗ 2. Создание класса и объекта в Java.	2	
3	Л 3. Принцип инкапсуляции, модификаторы доступа (private, public, protected).	1	
	ЛЗ 3. Применение инкапсуляции (сокрытие данных и методов).	2	
4	Л 4. Конструкторы, их виды (по умолчанию, параметризованные).	1	

5	ЛЗ 4. Создание объектов с использованием конструкторов.	2	
	Л 5. Принцип наследования, ключевое слово extends.	1	0
	ЛЗ 5. Применение наследования, создание базовых и производных классов.	2	15
	СРОП 1. Консультации по выполнению СРО 1 «Применение полиморфизма, перегрузка методов.»		0
МОДУЛЬ 2 Наследование, полиморфизм и абстракция.			
6	Л 6. Принцип полиморфизма, перегрузка и переопределение методов.	1	0
	ЛЗ 6. Реализация полиморфизма, использование методов override и overload.	2	15
	СРО 1. Прием СРС 1 в форме оцениваться по отчёту, демонстрации преекта.		20
7	Л 7. Принцип абстракции, абстрактные классы и интерфейсы.	1	0
	ЛЗ 7. Реализация абстракции в Java.	2	15
	СРОП 2. Консультации по выполнению СРО 2 «Абстракция и интерфейсы»		0
8	Л 8. Обработка исключений в Java, ключевые слова try, catch, finally.	1	0
	ЛЗ 8. Работа с исключениями, обработка ошибок.	2	15
	СРО 2 Прием СРС 2 в форме оцениваться по отчёту, демонстрации проекта.		20
Рубежный контроль 1			100
9	Л 9. Основы работы с коллекциями: List, Set, Map.	1	0
	ЛЗ 9. Использование коллекций для хранения данных.	2	10
10	Л 10. Чтение и запись файлов в Java, использование потоков.	1	0
	ЛЗ 10. Операции с файлами в Java (чтение и запись).	2	10
МОДУЛЬ 3 Исключения, коллекции и работа с базами данных			
11	Л 11. Основы работы с базами данных через JDBC.	1	0
	ЛЗ 11. Подключение к базе данных с использованием JDBC.	2	10
	СРОП 3. Консультации по выполнению СРО 3 «Выполнение операций с базой данных, выполнение SQL-запросов»		0
12	Л12. Основы многозадачности, создание потоков в Java.	1	0
	ЛЗ 12. Создание и управление потоками.	2	10
	СРО 3. Прием СРС 3 в форме оцениваться по отчёту, демонстрации проекта.		15
13	Л 13. Основы проектирования с использованием UML-диаграмм.	1	0
	ЛЗ 13. Создание UML-диаграмм для проектов на Java.	2	10
	СРОП 4. Консультации по выполнению СРО 4 «Завершающая самостоятельная работа над проектом с использованием ООП».		0
14	Л 14. Итоги курса, обзор основных концепций и принципов.	1	0
	ЛЗ 14. Разработка итогового проекта, использование всех изученных технологий.	2	10
15	Л 15. Презентация итогового проекта, обсуждение.	1	0
	ЛЗ 15. Студенты представляют свои проекты, используя все изученные принципы ООП.	2	10
	СРО 4. Прием СРС 4 в форме оцениваться по отчёту, демонстрации проекта.		15
Рубежный контроль 2			100
Итоговый контроль (экзамен)			100
ИТОГО за дисциплину			100

РУБРИКАТОР СУММАТИВНОГО ОЦЕНИВАНИЯ КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

СРС 1. «Применение полиморфизма, перегрузка методов» (20% от 100% РК1)

Критерий	«Отлично» 15-20 %	«Хорошо» 9-14%	«Удовлетворительно» 5-8%	«Неудовлетворительно» 2-4%
Теоретическая часть	Все основные принципы полиморфизма и перегрузки методов полностью и точно объяснены. Концепции подкреплены конкретными примерами из Java.	Основные принципы полиморфизма и перегрузки методов объяснены, но некоторые аспекты не полностью раскрыты или примеры недостаточны.	Принципы полиморфизма и перегрузки методов объяснены в общих чертах, примеры недостаточно ясны.	Теоретическая часть неполная, основные принципы полиморфизма и перегрузки методов не раскрыты.
Практическая часть	Полиморфизм и перегрузка методов полностью реализованы с использованием конкретных примеров. Код без ошибок, применены все аспекты полиморфизма и перегрузки.	Реализованы основные аспекты полиморфизма и перегрузки методов, но часть кода неполная или с ошибками.	Реализованы только основные аспекты полиморфизма и перегрузки методов, присутствуют ошибки в коде, некоторые части не работают.	Полиморфизм и перегрузка методов не применяются должным образом или отсутствуют, код содержит множество ошибок.
Структура проекта и проектирование	Структура проекта ясна и логична. Полиморфизм и перегрузка методов используются эффективно, код читаем и хорошо организован.	Структура проекта понятна, но применение полиморфизма и перегрузки методов не всегда оптимально или есть избыточные элементы.	Структура проекта сложная, но из-за неправильного использования полиморфизма и перегрузки методов код трудно понять.	Структура проекта неправильная, полиморфизм и перегрузка методов не применяются должным образом, проект неэффективен.

СРС 2. «Абстракция и интерфейсы» (20% от 100% РК1)

Критерий	«Отлично» 15-20%	«Хорошо» 9-14%	«Удовлетворительно» 5-8%	«Неудовлетворительно» 2-4%
Теоретическая часть	Все основные принципы абстракции и интерфейсов полностью и точно объяснены. Четко разъяснены различия между абстрактными классами и интерфейсами, их применение в Java с примерами.	Принципы абстракции и интерфейсов объяснены, но некоторые аспекты не полностью раскрыты или примеры недостаточны.	Принципы абстракции и интерфейсов объяснены в общих чертах, но примеры неполные или не совсем точные.	Теоретическая часть неполная, основные принципы абстракции и интерфейсов не раскрыты.
Практическая часть	Абстракция и интерфейсы реализованы на высоком уровне	Реализованы основные концепции абстракции и интерфейсов, но в	Абстракция и интерфейсы применяются только частично,	Применение абстракции и интерфейсов отсутствует или

	использованием Java. Примеры кода полноценные и корректно работают, решение задачи эффективно.	некоторых местах код неполный или с ошибками.	имеются ошибки в реализации, код не всегда работает корректно.	неправильно реализовано. Код содержит ошибки, функциональность ограничена.
Структура проекта и проектирование	Структура проекта логична и ясна. Абстракция и интерфейсы используются эффективно для улучшения архитектуры приложения, код легко читается и расширяется.	Структура проекта понятна, но абстракция и интерфейсы используются не везде, где это возможно, или их использование не всегда эффективно.	Структура проекта сложная, но не оптимизирована из-за неправильного использования абстракции и интерфейсов, код трудно читать.	Структура проекта неправильная, абстракция и интерфейсы не используются или использованы неэффективно.

СРС 3. «Выполнение операций с базой данных, выполнение SQL-запросов» (15% от 100% РК2)

Критерий	«Отлично» 10-15 %	«Хорошо» 6-9%	«Удовлетворительно» 3-5%	«Неудовлетворительно» 0-2%
Теоретическая часть	Все основные операции с базой данных (вставка, обновление, удаление, выборка) и SQL-запросы подробно объяснены. Теория подкреплена примерами выполнения различных SQL-запросов, включая JOIN, группировки и подзапросы.	Основные операции с базой данных и SQL-запросы объяснены, но некоторые аспекты, такие как сложные JOIN или подзапросы, недостаточно раскрыты или примеры неполные.	Основные операции с базами данных объяснены, но использование сложных SQL-запросов и их оптимизация не рассматриваются. Примеры SQL-запросов базовые.	Теоретическая часть неполная или слишком общая, не объяснены сложные запросы, примеры SQL-запросов отсутствуют или ошибочны.
Практическая часть	Все операции с базой данных (вставка, обновление, удаление, выборка) реализованы с использованием SQL-запросов, включая сложные запросы (JOIN, подзапросы, группировки). Запросы работают корректно и эффективно.	Основные операции с базой данных реализованы с помощью SQL-запросов, но сложные запросы могут работать с ошибками или не оптимально.	Реализованы только базовые SQL-запросы (SELECT, INSERT, UPDATE), но они могут содержать ошибки или не работать должным образом для более сложных операций.	SQL-запросы не реализованы или работают с ошибками, операции с базой данных не выполняются должным образом.
Структура проекта и проектирование	Структура базы данных логична и эффективна, все таблицы правильно нормализованы, запросы оптимизированы. Строение базы данных и ее связи понятны и соответствуют бизнес-логике приложения.	Структура базы данных правильная, но нормализация не полностью выполнена или некоторые связи между таблицами не оптимальны.	Структура базы данных не полностью нормализована, некоторые таблицы или связи между ними могут быть избыточными или избыточными.	Структура базы данных неправильная, отсутствует нормализация или связи между таблицами не имеют логики.

СРС 4. «Завершающая самостоятельная работа над проектом с использованием ООП» (15% от 100% РК2)

Критерий	«Отлично» 10-15 %	«Хорошо» 6-9%	«Удовлетворительно» 3-5%	«Неудовлетворительно» 0-2%
Теоретическая часть	Применение принципов ООП (инкапсуляция, наследование, полиморфизм, абстракция) полностью раскрыто, в проекте использованы все ключевые концепции ООП. Принципы подкреплены примерами, четко объяснены и на практике применены.	Принципы ООП объяснены, но не все ключевые концепции использованы в проекте или объяснены неполно.	В проекте используются только базовые концепции ООП, принципы объяснены на примере простых задач.	В проекте отсутствует использование принципов ООП или они неправильно применены.
Практическая часть	Проект выполнен с использованием ООП, все объекты, классы и их взаимодействия логично спроектированы. Код читаемый, хорошо структурирован, а все методы и классы правильно инкапсулированы.	Проект выполнен с использованием ООП, но некоторые классы или методы могли бы быть спроектированы более эффективно. Логика работы объектов не всегда ясна.	В проекте используются только базовые принципы ООП. Некоторые части кода плохо структурированы или методы/классы неэффективно взаимодействуют.	Проект выполнен без использования ООП или использование ООП неполное. Код неструктурирован, сложен для понимания.
Структура и проектирование	Проект имеет четкую, логичную структуру. Все компоненты системы правильно разделены на классы, классы взаимодействуют через хорошо спроектированные интерфейсы и абстракции. Применены паттерны проектирования.	Структура проекта понятная, но в некоторых местах можно было бы улучшить взаимодействие между классами или оптимизировать архитектуру.	Структура проекта не совсем логична, классы плохо взаимодействуют, отсутствуют абстракции или интерфейсы, проект не расширяем.	Структура проекта не логична, классы не взаимодействуют должным образом. Проект не имеет четкой архитектуры, трудно расширять.

Декан Т.С. Иманкулов

Председатель Академического комитета
по качеству преподавания и обучения Ж.А. Бурibaев

Заведующий кафедрой А.Н.Шормакова

Лектор М.Б. Байкуевков