

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Узунов Федор Владимирович

Должность: Ректор

Дата подписания: 30.09.2026 09:20:02

Уникальный программный ключ:

fd935d10451b860e912264c037858448452bfdb603f94788008e29877a6bcb5f5

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
«ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ»
«УНИВЕРСИТЕТ ЭКОНОМИКИ И УПРАВЛЕНИЯ»
Факультет экономики, управления и юриспруденции
Кафедра «Управление и бизнес-информатика»**



УТВЕРЖДАЮ

Проректор по учебно-

методической работе

_____/ Г.П. Узунова

30.09.2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Технологии программирования

Направление подготовки

10.03.01 «Информационная безопасность»

Профиль

Безопасность компьютерных систем

Квалификация выпускника

Бакалавр

Для всех

форм обучения

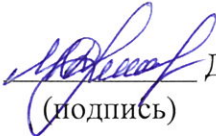
Симферополь, 2026

Рабочая программа составлена в соответствии с ФГОС ВО – бакалавриат по направлению подготовки 10.03.01 «Информационная безопасность» профиль Безопасность компьютерных систем, утвержденным приказом Министерства образования и науки Российской Федерации от 17.11.2020 г. № 1427 (зарегистрировано в Министерстве юстиции РФ 18.02.2021 №62548) с изменениями и дополнениями.

Программу составил Л. В. Яковенко, старший преподаватель

Рабочая программа дисциплины «Технологии программирования» утверждена на заседании кафедры «Управление и бизнес-информатика».

Протокол № 6 от 29.01.2026 г.

Заведующий кафедрой  Д.В. Моторина
(подпись)

АННОТАЦИЯ	
Индекс дисциплины по учебному плану	Наименование дисциплины
Б1.О.26	Технологии программирования
Цель изучения дисциплины	сформировать знания в области информационных технологий, выработать необходимые умения и навыки использования современных аппаратных и программных средств сбора, представления, хранения, передачи, обработки, анализа данных в профессиональной деятельности.
Место дисциплины в структуре ОПОП	Дисциплина относится к обязательной части блока 1. «Дисциплины (модули)» программы бакалавриата
Компетенции, формируемые в результате освоения дисциплины	ОПК-7
Содержание дисциплины	Тема 1. Объектно-ориентированное программирование (ООП): инкапсуляция, наследование и полиморфизм Тема 2. Парадигмы функционального программирования: чистые функции и неизменяемое состояние Тема 3. Паттерны проектирования: структурные и поведенческие шаблоны (Gang of Four) Тема 4. Микросервисная архитектура против монолитной: принципы построения распределенных систем Тема 5. Контейнеризация и оркестрация микросервисов: Docker и Kubernetes . Тема 6. Непрерывная интеграция и доставка (CI/CD): автоматизация жизненного цикла ПО
Общая трудоемкость дисциплины	Общая трудоемкость дисциплины составляет 6 зачетные единицы (216часов)
Форма промежуточной аттестации	Экзамен

Содержание

1. Цель и перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения программы бакалавриата	5
2. Место дисциплины в структуре ОПОП бакалавриата	5
3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся	5
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий	6
5. Контроль качества освоения дисциплины	11
6. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	11
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	12
8. Методические указания для обучающихся по освоению дисциплины	13
9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)	13
10. Описание материально-технического обеспечения, необходимого для осуществления образовательного процесса по дисциплине	14
Приложение к РПД	15

1. Цель и перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения программы бакалавриата

Цель изучения дисциплины «Технологии программирования» – сформировать знания в области информационных технологий, выработать необходимые умения и навыки использования современных аппаратных и программных средств сбора, представления, хранения, передачи, обработки, анализа данных в профессиональной деятельности.

В результате освоения ОПОП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

Коды компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ОПК-7	Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности	ОПК-7.1. Выбирает структуры данных и разрабатывает алгоритмы решения задач профессиональной деятельности ОПК-7.2. Выполняет отладку и тестирование программ ОПК-7.3. Разрабатывает и реализовывает на языке высокого уровня алгоритмы решения профессиональных задач

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина Б1.О.26 «Технологии программирования» относится к обязательной части блока 1. «Дисциплины (модули)» учебного плана ОПОП бакалавриата по направлению подготовки 10.03.02 «Информационная безопасность». Дисциплина «Информационные технологии в профессиональной деятельности» изучается обучающимися очной формы обучения в 5 семестре, очно-заочной формы обучения – в 5 семестре.

Знания, умения и навыки, полученные при изучении дисциплины «Технологии программирования», будут необходимы для углубленного и осмысленного восприятия дисциплин: Информатика и основы программирования, Операционные системы и др.

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 6 зачетных единицы (з.е.), 216 академических часа.

3.1. Объём дисциплины по видам учебных занятий (в часах)

Для очной формы обучения

Общая трудоёмкость дисциплины составляет 6 зачётных единицы 216 часа

Объём дисциплины	Всего часов
Общая трудоемкость дисциплины	216
Контактная работа	52

Аудиторная работа (всего):	52
Лекции	18
Семинары, практические занятия	34
Самостоятельная работа обучающихся (всего)	128
Курсовая работа	-
Зачет	-
Экзамен	36

Для очно-заочной формы обучения

Общая трудоёмкость дисциплины составляет 6 зачётных единиц 216 часа

Объём дисциплины	Всего часов
Общая трудоёмкость дисциплины	216
Контактная работа	40
Аудиторная работа (всего):	40
Лекции	20
Семинары, практические занятия	20
Самостоятельная работа обучающихся (всего)	140
Курсовая работа	-
Зачет	-
Экзамен	36

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины и трудоёмкость по видам учебных занятий (в академических часах)

№ темы	Наименование темы	Всего		Количество часов					
		ОФО	ОЗФО	Контактная работа				Внеаудит. работа	
				Лекции		Практические		Самост. работа	
				ОФО	ОЗФО	ОФО	ОЗФО	ОФО	ОЗФО
1.	Тема 1. Объектно-ориентированное программирование (ООП): инкапсуляция, наследование и полиморфизм	34	34	2	2	4	2	28	30
2.	Тема 2. Парадигмы функционального программирования: чистые функции и неизменяемое состояние	34	34	4	2	4	2	28	30
3.	Тема 3. Паттерны	34	34	4	2	4	2	28	30

	проектирования: структурные и поведенческие шаблоны (Gang of Four)								
4.	Тема 4. Микросервисная архитектура против монолитной: принципы построения распределенных систем	34	32	4	1	4	1	28	30
5.	Тема 5. Контейнеризация и оркестрация микросервисов: Docker и Kubernetes .	32	32	4	1	4	1	28	30
6.	Тема 6. Непрерывная интеграция и доставка (CI/CD): автоматизация жизненного цикла ПО	36	30	6	2	6	2	24	26
	Всего по дисциплине	180	180	18	20	34	20	128	140
	Контроль	36	36						
	Итого	216	216						

4.2. Содержание дисциплины, структурированное по темам (разделам)

Тема 1. Объектно-ориентированное программирование (ООП): инкапсуляция, наследование и полиморфизм Раскрывается парадигма программирования, в которой основным строительным блоком является объект — структура, объединяющая данные (поля/свойства) и методы работы с ними.

Инкапсуляция: сокрытие внутренней реализации объекта и защита его состояния от несанкционированного доступа извне через модификаторы видимости (*private*, *protected*, *public*) и интерфейсы.

Наследование: механизм создания иерархий классов, позволяющий дочерним классам расширять или переопределять функциональность родительских, что способствует повторному использованию кода.

Полиморфизм: способность объектов с одинаковым интерфейсом иметь разную реализацию (например, виртуальные функции и позднее связывание). Рассматривается на примере классических паттернов проектирования (Стратегия, Фабричный метод).

Тема 2. Парадигмы функционального программирования: чистые функции и неизменяемое состояние Анализируется подход к разработке, исключающий изменение данных после их создания.

Раскрываются понятия чистых функций (результат зависит только от входных аргументов, отсутствуют побочные эффекты) и функций высшего порядка.

Изучается работа со списками через операции *map* (преобразование), *filter* (фильтрация) и *reduce/fold* (агрегация).

Рассматриваются замыкания (*closures*) и каррирование. Практическая часть включает работу с лямбда-выражениями в современных императивных языках (Java Streams API, Python *lambdas*, C# LINQ) и применение этих принципов для написания безопасного многопоточного кода.

Тема 3. Паттерны проектирования: структурные и поведенческие шаблоны (Gang of Four) Тема посвящена типовым архитектурным решениям часто возникающих задач при проектировании программного обеспечения.

Структурные паттерны: Адаптер (согласование несовместимых интерфейсов), Декоратор (динамическое добавление обязанностей объектам), Фасад (предоставление простого интерфейса к сложной подсистеме).

Поведенческие паттерны: Наблюдатель (*Observer*, основа событийно-ориентированных систем), Стратегия (инкапсуляция взаимозаменяемых алгоритмов), Команда (параметризация действий объектами). Для каждого паттерна раскрывается UML-диаграмма структуры, мотивация применения и пример реализации.

Тема 4. Микросервисная архитектура против монолитной: принципы построения распределенных систем Сравниваются два фундаментальных подхода к архитектуре серверных приложений.

Монолит: единая кодовая база, централизованное развертывание, тесная связанность компонентов. Плюсы — простота разработки на старте; минусы — сложность масштабирования и технологического стека.

Микросервисы: приложение разбивается на набор мелких, независимо деплолируемых сервисов, каждый из которых отвечает за свою бизнес-область и имеет собственную базу данных. Раскрываются механизмы межсервисного взаимодействия (*REST/gRPC*, брокеры сообщений типа *RabbitMQ/Kafka*), использование *Service Discovery* и *API Gateway*. Анализируются проблемы распределенных транзакций (*Saga*-паттерн) и консистентности данных.

Тема 5. Контейнеризация и оркестрация микросервисов: Docker и Kubernetes Рассматриваются технологии упаковки и управления приложениями в промышленной эксплуатации.

Docker: изучается устройство контейнера как легковесной среды изоляции на уровне ОС (*cgroups*, *namespaces*), написание файлов конфигурации (*Dockerfile*) и управление образами через реестры.

Kubernetes (*K8s*): раскрытие архитектуры системы оркестрации. Понятия пода (*Pod*), сервиса (*Service*), контроллера развертывания (*Deployment*) и хранилища (*PersistentVolume*). Описываются процессы автоматического самовосстановления, масштабирования нагрузки (*Horizontal Pod Autoscaler*) и проведения обновлений без простоя (*Rolling Update*).

Тема 6. Непрерывная интеграция и доставка (CI/CD): автоматизация жизненного цикла ПО Раскрывается современный стандарт командной разработки, минимизирующий ручной труд при сборке, тестировании и развертывании продукта.

Непрерывная интеграция (*CI*): автоматический запуск сборки и модульных тестов при каждом слиянии кода в общий репозиторий (использование инструментов *Jenkins*, *GitLab CI*, *GitHub Actions*).

Непрерывная доставка/развертывание (*CD*): автоматический прогон кода через стадии тестирования (*QA*, нагрузочные тесты) и выкатка в продуктивную среду.

Рассматривается концепция «инфраструктура как код» (*IaC*) с помощью инструментов *Terraform* и *Ansible* для автоматизации настройки серверного окружения.

4.3. Содержание практических занятий (очная форма обучения)

Тема 1. Объектно-ориентированное программирование (ООП): инкапсуляция, наследование и полиморфизм Раскрывается парадигма программирования, в которой основным строительным блоком является объект — структура, объединяющая данные (поля/свойства) и методы работы с ними.

Инкапсуляция: сокрытие внутренней реализации объекта и защита его состояния от несанкционированного доступа извне через модификаторы видимости (*private*,

<i>protected, public</i>) и интерфейсы. <i>Наследование</i>: механизм создания иерархий классов, позволяющий дочерним классам расширять или переопределять функциональность родительских, что способствует повторному использованию кода.
Тема 2. Парадигмы функционального программирования: чистые функции и неизменяемое состояние Анализируется подход к разработке, исключающий изменение данных после их создания. Раскрываются понятия чистых функций (результат зависит только от входных аргументов, отсутствуют побочные эффекты) и функций высшего порядка. Изучается работа со списками через операции <i>map</i> (преобразование), <i>filter</i> (фильтрация) и <i>reduce/fold</i> (агрегация).
Тема 3. Паттерны проектирования: структурные и поведенческие шаблоны (Gang of Four) Тема посвящена типовым архитектурным решениям часто возникающих задач при проектировании программного обеспечения. Структурные паттерны: Адаптер (согласование несовместимых интерфейсов), Декоратор (динамическое добавление обязанностей объектам), Фасад (предоставление простого интерфейса к сложной подсистеме).
Тема 4. Микросервисная архитектура против монолитной: принципы построения распределенных систем Сравниваются два фундаментальных подхода к архитектуре серверных приложений. Монолит: единая кодовая база, централизованное развертывание, тесная связанность компонентов. Плюсы — простота разработки на старте; минусы — сложность масштабирования и технологического стека.
Тема 5. Контейнеризация и оркестрация микросервисов: Docker и Kubernetes Рассматриваются технологии упаковки и управления приложениями в промышленной эксплуатации. Docker: изучается устройство контейнера как легковесной среды изоляции на уровне ОС (cgroups, namespaces), написание файлов конфигурации (<i>Dockerfile</i>) и управление образами через реестры.
Тема 6. Непрерывная интеграция и доставка (CI/CD): автоматизация жизненного цикла ПО Раскрывается современный стандарт командной разработки, минимизирующий ручной труд при сборке, тестировании и развертывании продукта. Непрерывная интеграция (CI): автоматический запуск сборки и модульных тестов при каждом слиянии кода в общий репозиторий (использование инструментов Jenkins, GitLab CI, GitHub Actions). Непрерывная доставка/развертывание (CD): автоматический прогон кода через стадии тестирования (QA, нагрузочные тесты) и выкатка в продуктивную среду.

4.4. Содержание самостоятельной работы

Тема 1. Объектно-ориентированное программирование (ООП): инкапсуляция, наследование и полиморфизм <i>Полиморфизм</i>: способность объектов с одинаковым интерфейсом иметь разную реализацию (например, виртуальные функции и позднее связывание). Рассматривается на примере классических паттернов проектирования (Стратегия, Фабричный метод).
Тема 2. Парадигмы функционального программирования: чистые функции и неизменяемое состояние Рассматриваются замыкания (<i>closures</i>) и каррирование. Практическая часть включает работу с лямбда-выражениями в современных императивных языках (Java Streams API, Python lambdas, C# LINQ) и применение этих

принципов для написания безопасного многопоточного кода. Тема 3. Паттерны проектирования: структурные и поведенческие шаблоны (Gang of Four) Поведенческие паттерны: Наблюдатель (<i>Observer</i>, основа событийно-ориентированных систем), Стратегия (инкапсуляция взаимозаменяемых алгоритмов), Команда (параметризация действий объектами). Для каждого паттерна раскрывается UML-диаграмма структуры, мотивация применения и пример реализации. Тема 4. Микросервисная архитектура против монолитной: принципы построения распределенных систем Микросервисы: приложение разбивается на набор мелких, независимо деплолируемых сервисов, каждый из которых отвечает за свою бизнес-область и имеет собственную базу данных. Раскрываются механизмы межсервисного взаимодействия (<i>REST/gRPC</i>, брокеры сообщений типа <i>RabbitMQ/Kafka</i>), использование <i>Service Discovery</i> и <i>API Gateway</i>. Анализируются проблемы распределенных транзакций (<i>Saga</i>-паттерн) и консистентности данных Тема 5. Контейнеризация и оркестрация микросервисов: Docker и Kubernetes <i>Kubernetes (K8s)</i>: раскрытие архитектуры системы оркестрации. Понятия пода (<i>Pod</i>), сервиса (<i>Service</i>), контроллера развертывания (<i>Deployment</i>) и хранилища (<i>PersistentVolume</i>). Описываются процессы автоматического самовосстановления, масштабирования нагрузки (<i>Horizontal Pod Autoscaler</i>) и проведения обновлений без простоя (<i>Rolling Update</i>). Тема 6. Непрерывная интеграция и доставка (CI/CD): автоматизация жизненного цикла ПО Рассматривается концепция «инфраструктура как код» (<i>IaC</i>) с помощью инструментов <i>Terraform</i> и <i>Ansible</i> для автоматизации настройки серверного окружения.

5. Контроль качества освоения дисциплины

Текущий контроль и промежуточная аттестация осуществляются в соответствии с «Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся в Автономной некоммерческой организации «Образовательная организация высшего образования» «Университет экономики и управления».

Вид промежуточной аттестации – экзамен. Форма проведения промежуточной аттестации – письменный экзамен.

Фонд оценочных средств по дисциплине приведен в приложении к РПД.

6. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная литература:

1. Терехов, А. Н. Технология программирования : учебное пособие / А. Н. Терехов. — 6-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2026. — 148 с. — ISBN 978-5-4497-3349-8. — Текст : электронный // Образовательная платформа IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/books/162544/details> (дата обращения: 27.01.2026). — Режим доступа: для авторизир. пользователей
2. Згуральская, Е. Н. Технологии программирования : учебное пособие / Е. Н. Згуральская. — Ульяновск : Ульяновский государственный технический университет, 2025. — 60 с. — ISBN 978-5-9795-2489-4. — Текст : электронный // Образовательная

платформа IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/books/163127/details> (дата обращения: 27.01.2026). — Режим доступа: для авторизир. пользователей

б) дополнительная литература:

3. Баженова, И. Ю. Введение в программирование : учебное пособие / И. Ю. Баженова, В. А. Сухомлин. — 5-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2026. — 326 с. — ISBN 978-5-4497-3298-9. — Текст : электронный // Образовательная платформа IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/books/162502/details> (дата обращения: 27.01.2026). — Режим доступа: для авторизир. пользователей

4. Основы информационных технологий : учебное пособие / С. В. Назаров, С. Н. Белоусова, И. А. Бессонова [и др.]. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 530 с. — ISBN 978-5-4497-0339-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/89454.html> (дата обращения: 16.01.2026). — Режим доступа: для авторизир. пользователей.

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Информационно-правовой портал «Гарант»: официальный сайт. — URL: <http://www.garant.ru> — Текст: электронный.

2. Цифровой образовательный ресурс «IPRsmart»: официальный сайт. — URL: <http://www.iprbookshop.ru/> — Текст: электронный.

3. Научная электронная библиотека «КиберЛенинка»: официальный сайт. — URL: <https://cyberleninka.ru/> — Текст: электронный.

4. Российский интернет-портал и аналитическое агентство TAdviser: официальный сайт. — URL: <https://www.tadviser.ru/> — Текст: электронный.

8. Методические указания для обучающихся по освоению дисциплины

При проведении лекций, семинарских (практических) занятий, самостоятельной работе обучающихся применяются интерактивные формы проведения занятий с целью погружения обучающихся в реальную атмосферу профессионального сотрудничества по разрешению проблем, оптимальной выработки навыков и качеств будущего специалиста. Интерактивные формы проведения занятий предполагают обучение в сотрудничестве. Все участники образовательного процесса (преподаватель и обучающиеся) взаимодействуют друг с другом, обмениваются информацией, совместно решают проблемы, моделируют ситуацию.

В учебном процессе используются интерактивные формы занятий:

- творческое задание. Выполнение творческих заданий требует от обучающегося воспроизведение полученной ранее информации в форме, определяемой преподавателем, и требующей творческого подхода;

- групповое обсуждение. Групповое обсуждение кого-либо вопроса направлено на достижение лучшего взаимопонимания и способствует лучшему усвоению изучаемого материала.

В ходе освоения дисциплины при проведении контактных занятий используются следующие формы обучения, способствующие формированию компетенций: лекции-

дискуссии; кейс-метод; решение задач; ситуационный анализ; обсуждение рефератов и докладов; разработка групповых проектов; встречи с представителями государственных и общественных организаций.

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

В процессе лекционных и практических занятий используется следующее программное обеспечение:

- *программы, обеспечивающие доступ в сеть «Интернет» (например, «Microsoft Edge», «Google Chrome»);
- *программы, демонстрации видео материалов (например, проигрыватель «Windows Media Player»);
- *текстовые редакторы и процессоры (например, «Microsoft Office Word»);
- *табличные процессоры (например, «Microsoft Office Excel»);
- *системы управления базами данных (например, «Microsoft Office Access»);
- *программы для демонстрации и создания презентаций (например, «Microsoft PowerPoint»);
- *проблемно-ориентированные пакеты прикладных программ по отраслям и сферам деятельности (например, «1С: Управление нашей фирмой», «Loginom Community Edition»).

10. Описание материально-технического обеспечения, необходимого для осуществления образовательного процесса по дисциплине

Учебная аудитория

Оборудование учебной аудитории:

рабочее место преподавателя; посадочные места по количеству обучающихся;
доска классная;
стенды информационные;

Учебно-наглядные пособия: компьютеры с лицензионным программным обеспечением и возможностью подключения к информационно-телекоммуникационной сети Интернет; мультимедийная установка.

Помещение для самостоятельной работы обучающихся

оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.