

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Структурное подразделение «Институт информационных технологий и анализа данных
(121)»

УТВЕРЖДЕНА:
на заседании совета института ИТиАД им. Е.И.Попова
Протокол №8 от 16 февраля 2026 г.

Рабочая программа дисциплины

«ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»

Направление: 09.03.02 Информационные системы и технологии

Информационные системы и технологии в административном управлении

Квалификация: Бакалавр

Форма обучения: очная

Документ подписан простой электронной подписью Составитель программы: Каташевцев Михаил Дмитриевич Дата подписания: 11.06.2026

Документ подписан простой электронной подписью Утвердил: Говорков Алексей Сергеевич Дата подписания: 14.06.2026

Документ подписан простой электронной подписью Согласовал: Аршинский Вадим Леонидович Дата подписания: 11.06.2026

Год набора – 2026

Иркутск, 2026 г.

1 Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

1.1 Дисциплина «Технологии программирования» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ПКС-2 Способен разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО	ПКС-2.3

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ПКС-2.3	Способен применять современные технологии программирования для разработки программного обеспечения	Знать основные технологии программирования и принципы разработки программного обеспечения. Уметь выполнять декомпозицию программного кода; писать «чистый код», разрабатывать приложения с графическим интерфейсом, выполнять тестирование и отладку разрабатываемого приложения с использованием современных технологий Владеть Современной интегрированной средой разработки (IDE), системами контроля версий, технологиями отладки и автоматического тестирования

2 Место дисциплины в структуре ООП

Изучение дисциплины «Технологии программирования» базируется на результатах освоения следующих дисциплин/практик: «Информатика», «Программирование на языке высокого уровня»

Дисциплина является предшествующей для дисциплин/практик: «Web-программирование», «Проектирование информационных систем»

3 Объем дисциплины

Объем дисциплины составляет – 4 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)	
	Всего	Семестр № 4
Общая трудоемкость дисциплины	144	144
Аудиторные занятия, в том числе:	48	48

лекции	16	16
лабораторные работы	32	32
практические/семинарские занятия	0	0
Самостоятельная работа (в т.ч. курсовое проектирование)	60	60
Трудоемкость промежуточной аттестации	36	36
Вид промежуточной аттестации (итогового контроля по дисциплине)	Экзамен	Экзамен

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Семестр № 4

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)				
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	
1	2	3	4	5	6	7	8	9	10	11
1	Практики написания “чистого кода”	1	2	1	6			2, 2	16	Отчет по лабораторной работе
2	Работа с системой контроля версий	2	2					3	2	Контрольная работа
3	Отладка	3	2					3	2	Контрольная работа
4	Автоматизация тестирования	4	2	2	4					Отчет по лабораторной работе
5	Реализация пользовательских типов в C Sharp	5	2	3	10			2	10	Отчет по лабораторной работе
6	Дженерики (обобщения)	6	2	4	6			3	2	Контрольная работа
7	Boxing и Unboxing	7	2					2	10	Отчет по лабораторной работе
8	Ленивые типы	8	2	5	6			1, 2	18	Отчет по лабораторной работе
	Промежуточная аттестация								36	Экзамен
	Всего		16		32				96	

4.2 Краткое содержание разделов и тем занятий

Семестр № 4

№	Тема	Краткое содержание
1	Практики написания “чистого кода”	Тема посвящена основным практикам написания чистого кода, таким как выбор наименования переменных, классов, файлов. Изучаются лучшие практики по оформлению качественного

		современного кода.
2	Работа с системой контроля версий	Тема посвящена основным приемам работы с системами контроля версий. Рассмотрены базовые операции push, pull, commit, checkout. Показываются методы создания веток, а также процесс слияния веток при работе в команде. Изучаем создание удалённого репозитория. Отдельно рассмотрена работа с git в IDE Visual Studio
3	Отладка	Тема посвящена методам отладки кода в современных IDE таких как Visual Studio. Узнаем что такое контексты переменных local, global, watch. Работа со стеком вызовов. Создание точек остановки.
4	Автоматизация тестирования	В теме рассмотрен вопрос автоматизированного тестирования приложения. Создание модульных, функциональных тестов. Отдельно изучаются возможности Visual Studio для работы с тестами.
5	Реализация пользовательских типов в C Sharp	На примере реализации математической сущности (фигура, матрица и т. д.) средствами языка C sharp изучается подход к созданию пользовательских типов. Отдельно изучаются особенности создания свойств, статических методов, объявление полей, переопределение арифметических и логических операторов функции equals
6	Дженерики (обобщения)	В теме изучаются особенности работы с общими типами (дженерики). Особенности реализации встроенных в язык контейнеров (списки, множества, словари). А также способы создания своих универсальных типов.
7	Boxing и Unboxing	В теме рассмотрена работа с полиморфными типами на языке C Sharp. Особенности работы с виртуальными и абстрактными функциями. Переопределение статических типов. Способы упаковки (boxing) и распаковки (unboxing) как полиморфных типов, так и типов не имеющих общих предков.
8	Ленивые типы	Тема посвящена работе с встроенными в C Sharp интерфейсами для реализации ленивых итераторов. Отдельно рассматривается применение подхода в обработки больших объемов данных. Отдельно рассмотрена работы с интерфейсом IComparable для создания универсальных сравнимых типов.

4.3 Перечень лабораторных работ

Семестр № 4

№	Наименование лабораторной работы	Кол-во академических часов
---	----------------------------------	----------------------------

1	Основы работы с ветвлениями, циклами и строками в CSharp	6
2	Разработка графического интерфейса на CSharp	4
3	Разработка пользовательского типа данных	10
4	Работа с дженериками в CSharp	6
5	Работа с графикой в CSharp	6

4.4 Перечень практических занятий

Практических занятий не предусмотрено

4.5 Самостоятельная работа

Семестр № 4

№	Вид СРС	Кол-во академических часов
1	Подготовка к контрольным работам	6
2	Подготовка к практическим занятиям (лабораторным работам)	48
3	Проработка разделов теоретического материала	6

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: обратная лекция

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по лабораторным работам:

<https://el.istu.edu/course/view.php?id=4022>

5.1.2 Методические указания для обучающихся по самостоятельной работе:

<https://el.istu.edu/course/view.php?id=4022>

6 Фонд оценочных средств для контроля текущей успеваемости и проведения промежуточной аттестации по дисциплине

6.1 Оценочные средства для проведения текущего контроля

6.1.1 семестр 4 | Отчет по лабораторной работе

Описание процедуры.

Отчет о сделанной работе фиксируется в README.md файле в git-репозитории

Критерии оценивания.

Отчет должен быть структурированным, информативным и отражать проделанную работу. Оценка зависит от полноты, ясности и технической корректности.

6.1.2 семестр 4 | Контрольная работа

Описание процедуры.

Часть 1. Работа с Git (5 баллов)

1. Теоретические вопросы (2 балла)

Что делает команда `git commit`? (1 балл)

Как создать новую ветку в Git и переключиться на неё? (1 балл)

2. Практическое задание (3 балла)

Создайте новый репозиторий на GitHub.

Добавьте файл `Program.cs` с кодом `Console.WriteLine("Hello, Git!");` и сделайте коммит.

Создайте ветку `feature`, измените текст на `Console.WriteLine("Hello, World!");` и сделайте коммит.

Смерджите ветку `feature` в `main` через `git merge`.

Часть 2. Отладка (Debugging) (5 баллов)

1. Теоретические вопросы (2 балла)

Что такое точка останова (breakpoint) и как её поставить в Visual Studio? (1 балл)

Как посмотреть значение переменной во время отладки? (1 балл)

2. Практическое задание (3 балла)

Дан код с ошибкой:

```
csharp
using System;

class Program
{
    static void Main()
    {
        int[] numbers = { 1, 2, 3, 4, 5 };
        Console.WriteLine("Sum: " + CalculateSum(numbers));
    }

    static int CalculateSum(int[] arr)
    {
        int sum = 0;
        for (int i = 0; i <= arr.Length; i++) // Ошибка здесь!
        {
            sum += arr[i];
        }
        return sum;
    }
}
```

Запустите отладку, найдите ошибку и объясните её. (1 балл)

Исправьте код и покажите пошаговое выполнение в режиме отладки. (2 балла)

Часть 3. Дженерики (Generics) (10 баллов)

1. Теоретические вопросы (2 балла)
- Зачем нужны дженерики в C#? Приведите пример. (1 балл)

Критерии оценивания.

- 17–20 баллов — отлично
 13–16 баллов — хорошо
 9–12 баллов — удовлетворительно

6.2 Оценочные средства для проведения промежуточной аттестации

6.2.1 Критерии и средства (методы) оценивания индикаторов достижения компетенции в рамках промежуточной аттестации

Индикатор достижения компетенции	Критерии оценивания	Средства (методы) оценивания промежуточной аттестации
ПКС-2.3	Демонстрирует знание основных принципов разработки программного обеспечения и базовых технологий программирования. Способен с использованием современной интегрированной среды разработки разрабатывать приложения с графическим интерфейсом, автоматизировать тестирование, выполнять отладку программного кода и работать с системой контроля версий.	Устное собеседование по теоретическим вопросам и выполнение практического задания

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Семестр 4, Типовые оценочные средства для проведения экзамена по дисциплине

6.2.2.1.1 Описание процедуры

Вопросы к экзамену:

1. Правил оформления кода в горизонтальной и вертикальной плоскости
2. Правила именования переменных, функций и классов.
3. Системы контроля версий, причины появления, применение в современной разработке
4. Основные команды git: pull, push, checkout, commit, clone. Примеры
5. Работа в команде с использованием git. Назначение веток. Привести пример использования при работе в команде.
6. Работы с удаленным репозиторием. Описать процесс подключения локального репозитория к удаленному.
7. Отладка. Назначение контрольных точек, стек вызовов, отслеживание изменения переменных.

8. Автоматическое. Тестирование. Назначение юнит-тестов, интеграционное тестирование.
9. Создание тестов в IDE, использование класса Assert, тестирование переменных и коллекций. Привести примеры.
10. Пользовательские типы, различие между структурой и классом. Примеры.
11. Особенности работы со структурой.
12. Передача переменных по ссылке и по значению. Примеры работы с функциями и списками.
13. Переопределение математических операторов при создании пользовательских типов. Примеры.
14. Переопределение логических операторов при создании пользовательских типов. Примеры.
15. Индексаторы и свойства в классах. Примеры
16. Переопределение функции equals, ее применение в тестировании. Пример переопределения функции equals
17. Переопределение ToString.
18. Использование \$-строк и @-строк
19. Дженерики. Основные классы и их назначения. Примеры
21. Класс List<T> и пример его реализации
22. Класс Set<T> и пример его реализации
23. Класс Dict<K, V> и пример его реализации
24. Назначение слова static, создание классов утилит. Примеры.
25. Переопределение статических функций. Пример.
26. Лямбда выражения, создание функций 1-го порядка. Примеры
27. Boxing и unboxing переменных
28. Основы LINQ, методы для работы со списками
29. Ленивые итераторы. Устройство IEnumerable
30. Пример реализации своего ленивого итератора
31. Интерфейс IComparable, пример реализации для своего типа
32. Применение IComparable при работы с LINQ

6.2.2.1.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительн о	Неудовлетворительно
глубокое понимание теоретических основ современных технологий программирования, уверенное владение современными инструментами программирования при решении типовых задач в рамках	хорошее понимание теоретических основ современных технологий программирования, уверенное владение современными инструментами программирования при решении типовых задач в рамках	Удовлетворительное понимание теоретических основ современных технологий программирования, базовое владение современными инструментами программирования при решении типовых задач в рамках предметной области, изложение ответов с ошибками	наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неуверенность и неточность ответов на дополнительные и наводящие вопросы, отсутствие навыков владения современными инструментами программирования

предметной области, грамотное и логически стройное изложение материала при ответах на все вопросы из билета, знание дополнительно рекомендованной литературы.	предметной области, грамотное и логически стройное изложение материала при ответах на все вопросы из билета	материала, уверенно исправляемыми после дополнительных вопросов, необходимость наводящих вопросов	
---	---	---	--

7 Основная учебная литература

1. Сосинская С. С. Использование языка С# в различных информационных технологиях : учебник для студентов вузов, обучающихся по направлению "Информационные системы и технологии" / С. С. Сосинская, 2014. - 367.

<http://elib.istu.edu/viewer/view.php?file=/files/er-2674.pdf>

[Сайт] – URL: -

2. Лаврищева Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева, 2020. - 432.

[Сайт] – URL: -

8 Дополнительная учебная литература и справочная

1. Паттерны проектирования / Э. Фримен, Э. Фримен при участии К. Сьерра и Б. Бейтса, 2015. - 645.

[Сайт] – URL: -

9 Ресурсы сети Интернет

1. <http://library.istu.edu/>
2. <https://e.lanbook.com/>
3. <https://github.com/>
4. <https://www.nuget.org>
5. <https://visualstudio.microsoft.com/ru/>
6. <https://stackoverflow.com>
7. <https://habr.com/>

10 Профессиональные базы данных

Не используются.

11 Перечень информационных технологий, лицензионных и свободно распространяемых специализированных программных средств, информационных справочных систем

1. Свободно распространяемое программное обеспечение Microsoft Visual Studio Community Edition (версия 2019 или выше)
2. Свободно распространяемое программное обеспечение git
3. Лицензионное программное обеспечение Системное программное обеспечение
4. Лицензионное программное обеспечение Пакет прикладных офисных программ
5. Лицензионное программное обеспечение Интернет-браузер

12 Материально-техническое обеспечение дисциплины

1. Учебная аудитория для проведения лекционных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Оснащение: комплект учебной мебели, рабочее место преподавателя, доска. Мультимедийное оборудование (в том числе переносное): мультимедийный проектор, экран, акустическая система, компьютер с выходом в интернет.
2. Учебная аудитория для проведения лабораторных/практических (семинарских) занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Оснащение: комплект учебной мебели, рабочее место преподавателя, доска. Мультимедийное оборудование (в том числе переносное): мультимедийный проектор, экран, акустическая система, компьютер с выходом в интернет.