

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ
РЕСПУБЛИКИ**

**Нарынский государственный университет им. С.Нааматова
Аграрно-технический факультет**

“СОГЛАСОВАНО”

Начальник учебного управления

Усубалиева Ж. усубалиева

« 5 » 09 2025 г.

“УТВЕРЖДАЮ”

Проректор по учебной работе

Омурова К. омурова

« 5 » 09 2025 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

По дисциплине: **«Технология программирования»**

Направление подготовки бакалавра: 710100 «Информатика и вычислительная техника»

Профиль подготовки: Автоматизированные системы обработки информации и управления

Форма обучения: очная

Нарын-2025г.

Рабочая учебная программа составлена на основе стандарта утвержденного МОиН КР. 21-сентября 2021 г. №1578/1 и учебного плана по данному направлению, утвержденному приказом НГУ им. С.Нааматова от 30.06.2022 г., протокол № 10/51

Рабочую программу составила п.и.к.  Бекежанов М.М.

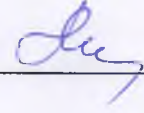
Рассмотрена и одобрена на заседании кафедры Информационные

технологии от « 3 » 09 2025 г., протокол № 1

Заведующий кафедрой, п.и.к.  У.У.Бейшеналиева

Рассмотрена и одобрена на заседании совета факультета от

« 4 » 09 2025 г., протокол № 2

Декан, доцент  А.Макеев

Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал, к.п.н. Бекежанов М.М. Кафедра информационных технологий, mbekezanov@gmail.com

1. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел " Б3.Б.8 Профессиональный" основной образовательной программы.

Дисциплина относится к базовой части профессионального цикла Б3 и изучается в 7го семестра 4-го курса.

2. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Зшифр компетенции	Расшифровка приобретаемой компетенции
ОК-13 (общекультурные компетенции)	владеет средствами самостоятельного, методически правильного использования методов физического воспитания и укрепления здоровья, готов к достижению должного уровня физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности

Шифр компетенции	Расшифровка приобретаемой компетенции
ОК-9 (общекультурные компетенции)	знание своих прав и обязанностей как гражданина своей страны; использование действующего законодательства, других правовых документов в своей деятельности; демонстрация готовности и стремления к совершенствованию и развитию общества на принципах гуманизма, свободы и демократии
ПК-13 (профессиональные компетенции)	способность разрабатывать средства автоматизированного проектирования информационных технологий
ПК-16 (профессиональные компетенции)	способностью учитывать аспекты корпоративной социальной ответственности при разработке и реализации стратегии организации
ПК-17 (профессиональные компетенции)	готовность проводить подготовку документации по менеджменту качества информационных технологий
ПК-18 (профессиональные компетенции)	способность использовать технологии разработки объектов профессиональной деятельности, в областях: машиностроение, приборостроение, наука, техника,
ПК-30 (профессиональные компетенции)	готовность проводить сборку информационной системы из готовых компонентов
ПК-5 (профессиональные компетенции)	способность проводить моделирование процессов и систем

В результате освоения дисциплины студент:

- должен знать: основы проектирования программ и алгоритмов;
- должен уметь: составлять алгоритмы и программы;
- должен владеть: владеть приемами структурного, объектно-ориентированного программирования;
- должен демонстрировать способность и готовность: выбора технологии и инструментальных средств, на их основе разработки, составления, отладки, тестирования и документирования программы на языках высокого уровня для задач обработки числовой и символьной информации.

Тематический план дисциплины

Лекции (16 занятий по 2 часа)

1. Введение в технологию программирования. Цикл разработки ПО. Ожидаемый результат: понимание основных этапов жизненного цикла программных средств.
2. Проектирование программных систем. Моделирование и алгоритмизация. Результат: умение создавать блок-схемы и алгоритмы решения.

3. Языки программирования: классификация, особенности. Результат: знание основных типов языков программирования.
4. Средства разработки программного обеспечения (IDE, компиляторы). Ожидается освоение интерфейса и функций основных средств.
5. Структуры данных и их использование. Результат: понимание основных структур и их применения.
6. Методы обработки ошибок и исключения. Результат: знание механизмов обработки ошибок.
7. Технологии разработки графического интерфейса. Результат: понимание принципов построения GUI.
8. Работа с базами данных в программировании. Результат: базовые знания по взаимодействию с БД.
9. Объектно-ориентированное программирование: концепции и принципы. Ожидается понимание ООП.
10. Проектирование классов и объектов. Результат: умение создавать классы и использовать их.
11. Тестирование программ и отладка. Результат: навыки отладки и тестирования.
12. Документирование программных продуктов. Результат: умение оформлять документацию.
13. Контроль версий и управление проектами. Результат: понимание систем контроля версий.
14. Автоматизация сборки и непрерывная интеграция. Результат: знакомство с современными инструментами.
15. Оптимизация и повышение производительности программ. Результат: методы оптимизации.
16. Итоги и обзор современных тенденций в технологии программирования. Результат: общее представление о современных подходах.

Лабораторные работы (16 занятий по 2 часа)

1. Установка и настройка среды разработки.
2. Создание и компиляция простых программ.
3. Практика алгоритмизации и работа с основными структурами данных.
4. Реализация обработки ошибок в программе.
5. Создание простого графического интерфейса.
6. Работа с файлами и базами данных.
7. Создание и использование классов.
8. Инкапсуляция и наследование.

9. Реализация полиморфизма.
10. Тестирование и отладка программ.
11. Документирование кода.
12. Работа с системами контроля версий.
13. Настройка инструментов автоматизации сборки.
14. Оптимизация кода на практике.
15. Групповой проект по разработке программного продукта.
16. Защита и презентация результатов лабораторных работ.

Образовательные технологии, включая интерактивные формы обучения

Основными образовательными технологиями проведения курса "Технологии программирования" являются:

- Лекции, сопровождаемые компьютерными презентациями;
- лабораторные работы, в рамках которых составляются и тестируются программы, иллюстрирующие теоретический материал лекций;
- самостоятельная работа студентов, включающая усвоение теоретического материала, поиск дополнительного материала и эффективных способов выполнения заданий, завершение выполнения лабораторных работ; оформление и подготовка к защите лабораторных работ, подготовка к текущему контролю знаний и к итоговому экзамену;

Примерные вопросы к экзамену:

1. Что такое технология программирования и какие задачи она решает?
2. Определение жизненного цикла программного обеспечения.
3. Основные модели жизненного цикла ПО (каскадная, спиральная и др.).
4. Что такое техническое задание на разработку программного обеспечения?
5. Понятие и виды алгоритмов, их свойства.
6. Основные структуры данных и способы их реализации.
7. Принципы структурного программирования.
8. Что такое процедурное программирование?
9. Основы объектно-ориентированного программирования (ООП): классы, объекты.
10. Инкапсуляция, наследование и полиморфизм в ООП.
11. Языки программирования: классификация и особенности.
12. Средства разработки программного обеспечения (IDE, компиляторы).
13. Основные элементы языка программирования C++ (или другого выбранного).

14. Методы обработки ошибок и исключения в программах.
15. Основы проектирования пользовательских интерфейсов.
16. Принципы работы с базами данных в программировании.
17. Что такое тестирование программ и основные методы отладки?
18. Документирование программного кода.
19. Системы контроля версий и их назначение.
20. Автоматизация сборки и непрерывная интеграция (CI/CD).
21. Методы оптимизации программного кода.
22. Критерии качества программного обеспечения.
23. Что такое модульность и компоненты ПО?
24. Основные стандарты и нормы в разработке ПО.
25. Понятие и виды спецификаций ПО.
26. Основы UML: диаграммы классов и последовательностей.
27. Принцип DRY и другие принципы чистого кода.
28. Среди языков программирования — отличия компилируемых и интерпретируемых.
29. Роль тестов в обеспечении надежности программного продукта.
30. Современные тенденции и новые технологии в сфере разработки ПО.

Тест

1. Что представляет собой технология программирования?
 - a) Совокупность методов и средств разработки ПО
 - b) Только язык программирования
 - c) Использование только аппаратных средств
 - d) Операционная система
2. Какой из этапов входит в жизненный цикл программного обеспечения?
 - a) Проектирование
 - b) Тестирование чипов
 - c) Монтаж электроники
 - d) Создание сайтов
3. Какая модель жизненного цикла чаще всего используется?
 - a) Каскадная
 - b) Линейная
 - c) Пирамидальная
 - d) Не применяется

4. Что такое алгоритм?
 - a) Четкое описание порядка действий для решения задачи
 - b) Часть операционной системы
 - c) Язык программирования
 - d) Текстовая документация
5. Какие свойства характерны для алгоритмов?
 - a) Дискретность и конечность
 - b) Неопределенность
 - c) Бесконечность
 - d) Низкая точность
6. Что относится к структурам данных?
 - a) Массивы, списки, деревья
 - b) Процессоры
 - c) Клавиатуры
 - d) Алфавиты языка
7. Что такое инкапсуляция в ООП?
 - a) Скрытие внутренней реализации объекта
 - b) Унаследование свойств от класса
 - c) Изменение кода
 - d) Тестирование ПО
8. Какой язык является объектно-ориентированным?
 - a) C++
 - b) HTML
 - c) SQL
 - d) CSS
9. Что такое компилятор?
 - a) Программа, преобразующая код в машинный язык
 - b) Аппаратный блок
 - c) Текстовый редактор
 - d) Система контроля версий
10. Что включает отладка программ?
 - a) Поиск и исправление ошибок
 - b) Документирование кода
 - c) Разработка дизайна
 - d) Подключение баз данных
11. Для чего нужен контроль версий?
 - a) Управление изменениями кода
 - b) Создание графического интерфейса
 - c) Проектирование базы данных
 - d) Оптимизация алгоритма

12. Что относится к методам оптимизации программ?
- a) Сокращение времени выполнения
 - b) Добавление комментариев
 - c) Увеличение объема кода
 - d) Использование устаревших библиотек
13. Что такое тестирование программ?
- a) Проверка правильности работы программ
 - b) Обновление ПО
 - c) Установка драйверов
 - d) Создание документации
14. Что такое модульность?
- a) Разделение программы на части
 - b) Увеличение скорости процессора
 - c) Упрощение интерфейса
 - d) Сжатие данных
15. Какой из языков является интерпретируемым?
- a) Python
 - b) C++
 - c) Assembler
 - d) Pascal
16. Что такое полиморфизм?
- a) Способность методов работать с разными типами данных
 - b) Создание пользовательского интерфейса
 - c) Сжатие файлов
 - d) Работа с базой данных
17. Что включает в себя документирование?
- a) Описание структуры и логики программы
 - b) Запуск программы
 - c) Оптимизацию кода
 - d) Тестирование
18. Что такое GUI?
- a) Графический интерфейс пользователя
 - b) Тип базы данных
 - c) Операционная система
 - d) Язык программирования
19. Какой из элементов не относится к структурам данных?
- a) Процессор
 - b) Дерево
 - c) Стек
 - d) Очередь

20. Что такое система непрерывной интеграции?
- a) Автоматизация сборки и тестирования кода
 - b) Резервное копирование данных
 - c) Разработка мобильных приложений
 - d) Безопасность сети
21. Основная цель использования баз данных в программировании?
- a) Хранение и управление данными
 - b) Создание графики
 - c) Разработка игр
 - d) Оптимизация кода
22. Какой из факторов влияет на качество ПО?
- a) Надежность
 - b) Вес компьютера
 - c) Скорость интернета
 - d) Количество клавиш на клавиатуре
23. Что такое наследование в ООП?
- a) Передача свойств и методов от родительского класса
 - b) Копирование кода вручную
 - c) Создание новой базы данных
 - d) Назначение прав доступа
24. Что такое алгоритмизация?
- a) Процесс разработки алгоритмов
 - b) Создание графики
 - c) Тестирование кода
 - d) Обновление программ
25. Что такое компиляция?
- a) Преобразование исходного кода в машинный
 - b) Тестирование программы
 - c) Создание документации
 - d) Установка драйверов
26. Что считается результатом выполнения программы?
- a) Выходные данные
 - b) Исходный код
 - c) Ошибки
 - d) Документация
27. Что такое объект в ООП?
- a) Экземпляр класса
 - b) Переменная
 - c) Процедура
 - d) Таблица базы данных

28. Зачем нужна автоматизация сборки?
- a) Ускорение создания конечного продукта
 - b) Изменение интерфейса
 - c) Создание дизайна
 - d) Ручная проверка
29. Что включает в себя процесс тестирования?
- a) Разработка тестов и их выполнение
 - b) Создание классов
 - c) Оптимизация кода
 - d) Написание документации
30. Что такое базовые принципы чистого кода?
- a) Понятность, простота, поддерживаемость
 - b) Сложность, многословность, длинный код
 - c) Отсутствие комментариев
 - d) Большое количество функций

Основная литература

1. Ларина Н.А. "Технологии программирования" — учебное пособие с теоретическим материалом, практическими заданиями и методикой выполнения курсовых работ.
2. Чарльз Петцольд "Код" — книга для понимания основ взаимодействия аппаратного обеспечения, операционных систем и программирования.
3. Марк Лутц "Python. Полное руководство" — практическое пособие по языку Python для начинающих программировать.
4. Скотт Майерс "Effective C++" — книга с углубленным изложением особенностей и лучших практик программирования на C++.
5. Василий Усов "Swift. Основы разработки приложений" — руководство по программированию на Swift для создания приложений под Apple.
6. Марейн Хавербеке "Выразительный JavaScript" — современное практическое руководство по JavaScript.
7. Рубцовский индустриальный институт. Учебное пособие "Технологии программирования".

Дополнительная литература

1. Мартин Фаулер «Рефакторинг. Улучшение проекта существующего кода» — методы улучшения и оптимизации кода через рефакторинг.
2. Эрик Фримен, Элизабет Робсон «Паттерны проектирования» — классические шаблоны для проектирования ПО с практическими примерами.
3. Александр Швец «Погружение в паттерны проектирования» — систематизированное представление о паттернах и их применении.
4. Дэвид Томас и Эндрю Хант «Программист-прагматик» — советы и стратегии профессионального роста программиста.
5. Антон Спрал «Думай как программист» — развитие нестандартного мышления и глубокое понимание языка C++.

6. Надежда Поликарпова, Анатолий Шалыто «Автоматное программирование» — теория конечных автоматов и автоматизация процессов разработки.

Интернет ресурсы

1. GeekBrains (geekbrains.ru) — онлайн-курсы по основам и продвинутому программированию на популярных языках с поддержкой наставников и практическими заданиями.
2. Яндекс.Практикум (practicum.yandex.ru) — интенсивные программы по программированию, предоставляющие опыт работы над реальными проектами и постоянную экспертную поддержку.
3. Открытое образование (openedu.ru) — бесплатные курсы от ведущих вузов России, включающие лекции, тесты и практические задания.
4. ITVDN (itvdn.com) — видеоуроки и курсы по современным IT-технологиям и языкам программирования с интерактивной практикой.
5. Skillbox (skillbox.ru) — широкий выбор курсов для разных уровней подготовки, включая введение в программирование и специализированные направления.
6. SkillFactory (skillfactory.ru) — курсы по программированию, веб-разработке, тестированию и кибербезопасности с интенсивной практикой.
7. BeOnMax (beonmax.com) — курсы по различным языкам программирования и веб-разработке, включая HTML, CSS, JavaScript, Python и другие.

Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Технология программирования" предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB, audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет.

Компьютерное оборудование имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

Для проведения дисциплины на кафедре имеются:

- Компьютерные классы, современные компьютеры которых объединены в локальную сеть.
- Имеется необходимое лицензионное программное обеспечение.
- Разработаны лабораторные работы, включающие в себя обучающие тексты, набор пошаговых инструкций, учебных задач и заданий, демонстрационный материал и тестовые задания.
- Лекционная аудитория оборудована проектором.