Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра автоматизированных систем управления
Кафедра автоматики
Кафедра вычислительной техники
Кафедра защиты информации

Паспорт экзамена

по дисциплине «Программирование», 2 семестр

Методика оценки

Экзамен проводится в устной форме по билетам. Билет состоит из 2 вопросов и формируется по следующему правилу:
первый вопрос выбирается из диапазона вопросов 1–24;
второй вопрос выбирается из диапазона вопросов 25–48.

Таким образом, проверяются результаты обучения по дисциплине, соотнесённые с установленными в программе индикаторами достижения компетенций ОПК-2.1, ОПК-2.2, ОПК-7.1, ОПК-7.2 и ОПК-7.3.

На экзамене преподаватель вправе задавать студенту уточняющие и дополнительные вопросы из общего перечня (п. 4).

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____
к экзамену по дисциплине «Программирование»

---

1. Вопрос 1
2. Вопрос 2.

Утверждаю: зав. кафедрой ____ _____ должность, ФИО
(подпись)
(дата)

Критерии оценки результатов обучения, соотнесенных с уровнями освоения индикаторов достижения компетенций

Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент проводит сравнительный комплексный анализ материала, выявляет проблемы, предлагает механизмы их решения, представляет количественные характеристики определенных процессов, приводит конкретные примеры, не допускает ошибок. Совокупность результатов обучения по дисциплине и соотнесенных с ними индикаторов достижения компетенций не содержит пробелов. Установленные в программе компетенции сформированы в полном объеме. Оценка составляет *от 35 до 40 баллов.*

Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, дает характеристику процессов, явлений, проводит анализ причин, условий, способен представить количественные и качественные характеристики процессов, не допускает существенных ошибок. Совокупность результатов обучения по дисциплине и соотнесенных с ними индикаторов достижения компетенций содержит несущественные пробелы. Установленные в программе компетенции сформированы на базовом уровне. Оценка составляет *от 27 до 34 баллов.*

Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, допускает непринципиальные ошибки. Совокупность результатов обучения по дисциплине и соотнесенных с ними индикаторов достижения компетенций содержит пробелы. Установленные в программе компетенции сформированы на пороговом уровне. Оценка составляет *от 20 до 26 баллов.*

Ответ на экзаменационный билет считается **неудовлетворительным** (ниже порогового уровня), если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, допускает принципиальные ошибки. Совокупность результатов обучения по дисциплине и соотнесенных с ними индикаторов достижения компетенций содержит существенные пробелы. Установленные в программе компетенции не сформированы. Оценка составляет *менее* 20 *баллов.*

Шкала оценки

Экзамен считается сданным, если сумма баллов по всем заданиям билета составляет от 20 до 40 баллов включительно. Сумма менее 20 баллов признается неудовлетворительным результатом промежуточной аттестации по дисциплине.

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, установленными в НГТУ.

Вопросы к экзамену **по дисциплине «**Программирование**»**

1. Понятие и виды программирования. Парадигмы и метафоры программирования; критерии качества программного обеспечения.
2. Основные принципы безопасного программирования: проверка входных данных, контроль границ и обработка ошибок.
3. Этапы работы программы. Формализация профессиональной задачи, входные и выходные данные, представление алгоритма.
4. Обоснование выбора языка и инструментальных средств программирования, в том числе отечественного производства. Компиляция и интерпретация.
5. Формирование исполняемой программы C++17: препроцессирование, компиляция, компоновка и запуск. Выполнение программы Python.

6. Работа с интегрированной средой разработки: создание проекта, сборка, диагностика ошибок и пошаговая отладка.
7. Базовые типы C++17 и Python: диапазоны значений, размер, представление и адресация данных.
8. Переменные, константы, выражения и преобразования типов. Переполнение и потеря точности.
9. Унарные, бинарные, тернарные и поразрядные операции. Приоритет операций; особенности инкремента и декремента.
10. Условные конструкции C++17 и Python. Оператор switch, перечисления enum и выбор способа ветвления.
11. Циклические конструкции C++17 и Python: синтаксис, блок-схемы, условия завершения и типичные ошибки.
12. Функции C++17 и Python: объявление, вызов, параметры и возвращаемые значения; способы передачи данных.
13. Рекурсивные функции: базовый случай, рекурсивный переход, стек вызовов и сравнение с итерацией.
14. Область видимости и время жизни переменных. Классы памяти и пространства имён.
15. Макросы C++: виды, назначение, преимущества и недостатки; сравнение с функциями и константами.
16. Организация многофайлового проекта C++17 и Python. Разделение интерфейса и реализации, заголовочные файлы и модули.
17. Модель памяти C++: стек, куча, размещение объектов и управление временем их жизни.
18. Статические одномерные массивы: объявление, инициализация, адресация и контроль границ.
19. Многомерные массивы C++ и списки Python: инициализация, обход и особенности адресации.
20. Указатели: объявление, инициализация, разыменование и нулевой указатель. Причины недопустимого доступа к памяти.
21. Операции и преобразования указателей. Связь указателей и массивов, допустимые пределы арифметики указателей.
22. Динамическая память: выделение и освобождение одномерных и многомерных массивов, утечки памяти и висячие указатели.
23. Структуры: объявление, инициализация, доступ к полям и передача в функции.
24. Линейный и двоичный поиск: условия применимости, организация данных и оценка числа операций.
25. Алгоритмы сортировки одномерного массива: выбором, вставками, обменами и быстрый метод; сравнение свойств.
26. Оценка сложности алгоритмов и обозначение «О-большое». Лучший, средний и худший случаи.
27. Амортизированная сложность: отличие от оценки отдельной операции, пример расширения динамического массива.
28. Библиотека algorithm: основные группы алгоритмов, работа с диапазонами, предикатами и лямбда-выражениями.
29. Основные правила оптимизации кода. Измерение производительности и сохранение корректности при оптимизации.

30. Обмен с потоками C++: ввод и вывод, форматирование, флаги состояния и обработка ошибок.
31. Буферизация выходных потоков, связанные потоки и влияние принудительного сброса буфера на производительность.
32. Файловые потоки: открытие, режимы доступа, чтение, запись, закрытие и проверка ошибок.
33. Строковые потоки: преобразование данных, разбор строки и форматирование результата.
34. Стандартная библиотека шаблонов STL. Последовательные и ассоциативные контейнеры, критерии выбора.
35. Контейнер vector: инициализация, доступ, изменение размера и вместимости; инвалидирование ссылок и итераторов.
36. Контейнеры map и set: назначение, поиск, вставка, удаление, уникальность ключей и трудоёмкость операций.
37. Контейнер vector<vector>: представление прямоугольных и ступенчатых данных, обход и изменение структуры.
38. Основные понятия криптографии. Открытый текст, шифротекст, ключ, шифрование и расшифрование; границы применения учебных реализаций.
39. Делимость целых чисел, наибольший общий делитель и наименьшее общее кратное. Алгоритм Евклида и его программная реализация.
40. Отношения на множестве, алгебраические структуры и кольцо целых чисел. Простые числа и их распределение.
41. Функция Эйлера, теоремы Эйлера и Ферма. Их применение при вычислениях по модулю.
42. Сравнения первой степени: условия разрешимости и способы программного решения.
43. Индексы, двучленные и показательные сравнения. Условия разрешимости сравнения x в степени n ≡ a (mod p).
44. Симметричные и асимметричные методы шифрования: принципы, различия и использование программных библиотек.
45. Механизмы сериализации и десериализации в языках высокого уровня. Сохранение структуры данных и проверка недоверенного ввода.
46. Бинарная сериализация: формат записи, порядок байтов, размер полей и проблемы переносимости.
47. JSON- и XML-сериализация: структура форматов, настройки, атрибуты и проверка восстановленных данных.
48. Дружественные функции C++: назначение, реализация и сравнение с доступом через get/set с точки зрения инкапсуляции и безопасности.