

Летняя сессия. Экзамен по дисциплине «Технологии программирования»

Экзамен состоит из 3 частей:

1. Тест. Тест состоит из 40 вопросов. Максимум – 8 баллов.
2. Теоретические вопросы. 8 вопросов, по вопросу на каждую тему. Задается случайный вопрос из темы, на который студент должен тут же ответить. Максимум – 8 баллов. Перечень вопросов приведен ниже.
3. Практическое задание. Задание, разбитое на 8 частей. Каждая часть оценивается в 2 балла. Язык программирования – C#. Максимум – 16 баллов. Выполняется последовательно, пропускать нельзя. Каждая часть принимается при условии ответа на вопрос по коду. В случае затруднения ответа дается время на «подумать». Всего дается 3 попытки. По окончании третьей попытки считаются выполненными только те части, по которым даны ответы по коду, не зависимо от того сколько частей было реализовано в коде. Перечень вопросов, для подготовки к практике, приведен ниже.

Сдавать все 3 части не обязательно. Каждый студент сам решает, что он будет сдавать, что нет, исходя из того, сколько баллов ему нужно набрать, но хотя бы что-то одно он должен сдать. При условии допуска студента до экзамена, он может учитывать работу в семестре (в таком случае считается сумма баллов, полученных в семестре и на экзамене), либо же нет.

- В первом случае оценка «5» начинается от 115,2 баллов и выше, оценка «4» от 96 баллов и до 115,2, оценка «3» от 76,8 баллов и до 96. Если студент набирает менее 76,8 баллов – оценка «2».
- Во втором случае оценка «5» начинается от 28,8 баллов и выше, оценка «4» от 24 баллов и до 28,8, оценка «3» от 16 баллов и до 24. Если студент набирает менее 16 баллов – оценка «2».

На пересдачах всегда действует только второй вариант, работа семестра не учитывается.

Также, правило при сдаче, сперва пишется тест, потом ответ на теоретические вопросы, затем выполнение практического задания. И только в таком порядке. Если студент отказывается от сдачи теории, потом он не может вернуться и ответить на нее, если ему не хватает баллов. Правило действует в рамках экзамена или одной пересдачи, на следующей пересдаче студент может снова выбирать из всех 3 вариантов сдачи.

Перечень теоретических вопросов по дисциплине «Технологии программирования»

1. Принципы программирования. Какие есть принципы программирования, в чем они заключаются?
2. Паттерны проектирования. На какие группы делятся паттерны проектирования? Примеры паттернов из каждой группы.
3. Архитектурные паттерны. Назначение архитектурных паттернов. Какие знаете?
4. Модули и библиотеки, модификатор `internal`. Что такое модульное программирование. Как работает модификатор `internal`.
5. IoC-контейнер. Зачем нужны и как работают IoC-контейнеры?
6. LINQ. Общие определения, где применяется.
7. Классический LINQ-запрос простой (`select`, `where`, `orderby`). Как строится? Как работает?
8. Классический LINQ-запрос сложный (`groupby`, `join`). Как строится? Как работает?
9. LINQ-запрос через методы расширения простые. Как строится? Как работает?
10. LINQ-запрос через методы расширения сложные. Как строится? Как работает?
11. ADO.NET. Как выполняется подключение к БД? Как создается и обрабатывают запросы к БД?
12. Linq to SQL. Как выполняется подключение к БД? Как создается и обрабатывают запросы к БД?

13. Entity Framework. Какие есть подходы к работе с БД?
14. ORM. Что это такое? Какие есть примеры?
15. Хранение данных. Какие типы хранения данных есть? Когда-какой применять?
16. MS Word. Что это такое, как с ним работать?
17. MS Excel. Что это такое, как с ним работать?
18. PDF. Что это такое, как с ним работать?
19. Работа с офисными пакетами. Как используются офисные пакеты при разработке программ?
20. Создание документа. Общие принципы создания документа/таблицы.
21. Сериализация. Что такое сериализация? Что такое десериализация? Область применения?
22. Подходы сериализации. Какие есть подходы сериализации?
23. Типы сериализации. Какие есть типы сериализации?
24. XML. Что такое XML? Каковы особенности записи?
25. JSON. Что такое JSON? Каковы особенности записи?
26. Потоки. Что такое поток? Что такое многопоточность?
27. Task. Зачем нужен класс Task? Как используются async/await?
28. Параллельное программирование. Зачем нужно? Какие подходы есть?
29. Синхронизация потоков. Зачем нужно синхронизировать потоки? Какие инструменты синхронизации потоков есть?
30. Потокобезопасные коллекции. Какие есть потокобезопасные коллекции?
31. Работа со строками. Какие есть особенности и подходы при обработке строк?
32. Форматирование. Какие типы как модно форматировать при выводе? Что такое CultureInfo?
33. Локализация. Зачем нужна локализация? Каков подход к локализации приложения?
34. Регулярные выражения. Назначение. Используемые символы, группы, привязки в выражениях
35. Регулярные выражения. Используемые кванторы, символы выбора. Примеры использования
36. Рефлексия. Что такое рефлексия? Какие основные элементы и классы в рефлексии?
37. Разбор типа. Как получить доступ к типу? Как получить доступ к элементам типа?
38. Создание объекта. Как создается объект через рефлексию?
39. Работа с объектом. Как считать данные из объекта типа через рефлексию? Как заполнить объект типа через рефлексию?
40. Атрибуты. Назначение атрибутов. Как описываются? Как применяются?

Перечень практических вопросов по дисциплине «Технологии программирования»

На практике будет задание, разбитое на 8 этапов. Каждый этап оценивается в 5 баллов

1. Как подключать библиотеку (свою или внешнюю) в другой проект? Как настраивается и применяется IoC-контейнеры? В чем заключается и как реализуется DAL-архитектура?
2. Как создавать LINQ-запрос?
3. Как задать таблицы для БД через EntityFramework? Как задать связи между таблицами? Как создать миграцию?
4. Где извлекаются данные для отчета? Как создавать pdf-файл? Как создавать в нем таблицу?
5. Как сохранять данные через сериализацию в форматах json и xml?
6. Как создавать и вызывать асинхронные методы? Как передавать туда параметры?
7. Как записывать регулярные выражения? Как их применять?
8. Как вызывать методы через рефлексию?