

Зимняя сессия. Экзамен по дисциплине «Разработка профессиональных приложений»

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 «Разработка профессиональных приложений»

1. Парадигмы программирования. Какие знаете парадигмы?
2. Концепции ООП. Какие есть концепции ООП. Что такое инкапсуляция? Что такое абстракция?
3. Классы и объекты. Что такое класс? Что может включаться в класс? Что такое объект?
4. Типы данных. Какие есть классификации? Что такое упаковка/распаковка?
5. Модификаторы доступа. Зачем нужны модификаторы доступа? Какие есть модификаторы доступа?
6. Конструкторы и деструкторы. Что такое конструктор? Как создается конструктор? Что такое деструктор? Как создается деструктор?
7. Наследование. Что такое наследование? Какие есть виды наследования? Как работает механизм преобразования типов? Как работает модификатор protected?
8. Наследование от базового класса. Какие есть особенности при наследовании от базового класса? Как можно переопределять методы базового класса?

9. Наследование от абстрактного класса. Какие есть особенности при наследовании от абстрактного класса? Зачем нужен модификатор `abstract`?
10. Наследование от интерфейса. Какие есть особенности при наследовании от интерфейса? Как наследоваться от нескольких интерфейсов?
11. Модификатор `static`. Какие особенности применения?
12. Методы расширения и Связывание методов. В чем заключается и как реализовать механизм расширения? Что такое связывание методов? Что такое раннее связывание? Что такое позднее связывание?
13. Полиморфизм. Что такое полиморфизм? Какие есть виды полиморфизма? Что такое полиморфизм подтипов?
14. Специализированный полиморфизм. В чем заключается специализированный полиморфизм? Какие правила перегрузки методов?
15. Параметрический полиморфизм. В чем заключается параметрический полиморфизм? Как создать объект от параметризованного класса? Как вызывать параметризованный метод?
16. Массивы. Как устроены массивы? Какие виды массивов есть?
17. Коллекции. На какие классификации делятся коллекции? В чем отличие коллекции от массива?
18. Индексируемые коллекции. В чем особенности индексируемых коллекций? Какие индексируемые коллекции знаете?
19. Ассоциативные коллекции. В чем особенности ассоциативных коллекций? Какие ассоциативные коллекции знаете?
20. Индексаторы и кортежи. Что такое индексатор? Как создать индексатор? Что такое кортеж? Где используется и как создается кортеж?
21. Делегаты. Что такое делегат? Что такое многоадресность?
22. Встроенные делегаты. Какие встроенные делегаты вы знаете? Где и как их использовать?
23. События. Что такое событие? Как создать событие? Чем событие отличается от делегата?
24. Анонимные методы. Что такое анонимный метод? Где и как применяется анонимный метод?
25. Лямбда-выражения. Что такое лямбда-выражение? Какие правила записи лямбда-выражения?
26. Потоки. Что такое поток? Какие операции можно проводить над потоком? Какие основные классы представлены в C# и в Java для работы с потоками?
27. Потоки. Методы чтения и записи. Какой основной метод записи данных в поток в C#, какие он принимает параметры? Какой основной метод чтения данных в поток в C#, какие он принимает параметры?
28. Потоки. Работа с файлами. Какие классы в C# реализуют абстракцию `Stream` для работы с файлами? Зачем нужны кодировки?
29. Потоки. Работа с памятью, сетью и обертки. Какова особенность работы с потоком памяти? Какова особенность работы с потоком данных из сети? Какие есть обертки?
30. Работа с файлами и каталогами. Какие основные классы в C# для работы с файлами? Какие методы они имеют? Какие основные классы в C# для работы с каталогами? Какие методы они имеют?
31. Исключения. Что такое исключение? Какие классификации ошибок и способов их обработок знаете? Что такое проброс исключения?
32. Оператор `try-catch`. Как устроен оператор `try-catch`? Каков механизм его работы?
33. Иерархия исключений. Что такое иерархия исключений? Как организовывается иерархия? Где применяется иерархия исключений?
34. Распространенные исключения. Какие часто встречаемые исключения можете перечислить? Что они означают? Как устранять подобные исключения?
35. Логирование действий. Зачем нужен логгер? Как устроен логгер?

36. Проверка на равенство. Где и как используется механизм проверки на равенство. Как реализовать механизм проверки на равенство для своих классов?
37. Сортировка. Где и как используется механизм сортировки. Как реализовать механизм сортировки для своих классов?
38. Создание дубликатов объекта. Какие особенности при создании копий объектов класса? Что такое поверхностное и глубокое копирование?
39. Итератор. Как устроен итератор? Где и как он применяется?
40. Ресурсы. Очистка ресурсов. Что такое управляемые и неуправляемые ресурсы? Зачем нужно очищать ресурсы? Что такое сборщик мусора? Как он работает?

Перечень практических вопросов по дисциплине «Разработка профессиональных приложений»

1. Как создавать классы, поля и методы классов? Как создавать объекты от классов? Как настроить доступ к элементам класса?
2. Как создавать полиморфный метод? Как создавать конструктор? Где и когда вызывается конструктор?
3. Как создавать интерфейсы? Как наследоваться от интерфейсов? Как создать индексатор?
4. Как создавать список? Как создавать список, содержащий в себе объекты разных классов? Как обращаться к элементам списка?
5. Как создавать события от встроенных делегатов? Как привязать/отвязать от события метод? Как вызвать событие?
6. Как создавать файл? Как записать данные в файл? Как сохранить файл?
7. Как создавать собственный класс ошибки? Как создавать выброс исключения? Как создавать обработку исключений?
8. Как реализовать механизм сортировки для пользовательских классов? Где и как используется методы, объявленные в применяемом стандартном интерфейсе? Какой альтернативный способ сортировки знаете?