1. **Паспорт Образовательной программы**

**«**Разработка программного обеспечения**»**

|  |  |
| --- | --- |
| **Версия программы** | **1** |
| **Дата Версии** | **14.10.2020** |

1. **Сведения о Провайдере**

|  |  |  |
| --- | --- | --- |
| 1.1 | Провайдер | государственное автономное профессиональное образовательное учреждение «Оренбургский колледж экономики и информатики» |
| 1.2 | Логотип образовательной организации | **D:\Работа\Учебный центр\Цифровые сертификаты\Логотип.jpg** |
| 1.3 | Провайдер ИНН | 5610046887 |
| 1.4 | Ответственный за программу ФИО | Асташева Татьяна Сергеевна |
| 1.5 | Ответственный должность | Преподаватель |
| 1.6 | Ответственный Телефон | +79198640281 |
| 1.7 | Ответственный Е-mail | astasheva.ts@mail.ru |

1. **Основные Данные**

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Описание** |
| 2.1 | Название программы | Разработка программного обеспечения |
| 2.2 | Ссылка на страницу программы | https://oksei.ru/page/personalnye\_cifrovye\_sertifikaty |
| 2.3 | Формат обучения | Онлайн |
|  | Подтверждение от ОО наличия возможности реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа | Программа реализуется посредством адаптивного обучающего комплекса «Ключевые компетенции цифровой экономики» на основе 1С-Битрикс: внутренний портал учебного заведения. |
| 2.4 | Уровень сложности | Базовый |
| 2.5 | Количество академических часов | 72 |
|  | Практикоориентированный характер образовательной программы: не менее 50 % трудоёмкости учебной деятельности отведено практическим занятиям и (или) выполнению практических заданий в режиме самостоятельной работы (кол-во академических часов) | Программа носит практикоориентированный характер: выполнению практических занятий отведено 60 % трудоемкости учебной деятельности; выполнению практических заданий в режиме самостоятельной работы – 4 часа. |
| 2.6 | Стоимость обучения одного обучающегося по образовательной программе, а также предоставление ссылок на 3 (три) аналогичные образовательные программы иных организаций, осуществляющих обучение, для оценки объективности стоимости или обоснование уникальности представленной образовательной программы в случае отсутствия аналогичных образовательных программ на рынке образовательных услуг | <https://itvdn.com/ru/live-online/net-developer> - 136 часов, 650$  <https://inordic.ru/courses/c-sharp> - 144 часа, 57 200 руб.  <https://geekbrains.ru/professions/microsoft_developer> - 110 часов, 80 000 руб. (48 000 ₽ при оплате до 31 октября 2020)  29 500 рублей |
| 2.7 | Минимальное количество человек на курсе | 25 |
| 2.8 | Максимальное количество человек на курсе | 100 |
| 2.9 | Данные о количестве слушателей, ранее успешно прошедших обучение по образовательной программе |  |
| 2.10 | Формы аттестации | Итоговая контрольная работа с выполнением кейса |
|  | Указание на область реализации компетенций цифровой экономики, к которой в большей степени относится образовательная программа, в соответствии с Перечнем областей | Программирование и создание ИТ-продуктов |

1. **Аннотация программы**

В результате прохождения обучения слушатели получат знания, умения и навыки, востребованные в профессиональной деятельности при программировании и создания программных продуктов.

Данная программа направлена на формирование у слушателей профессиональной компетенции:

ПК 1. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Под компетенцией понимается умение разрабатывать программное обеспечение с помощью платформы .NET, а также баз данных SQL Management Studio, работать с различными версиями проекта.

В результате освоения профессиональной компетенции будут получены знания по основные принципы технологии структурного и объектно-ориентированного программирования; этапы разработки программного обеспечения, а также основам проектирования баз данных; умения формировать алгоритмы разработки программных модулей в соответствии с техническим заданием, создавать программу по разработанному алгоритму как отдельный модуль. Слушатель овладеет навыками по разработке алгоритма решения поставленной задачи и реализовывать его средствами автоматизированного проектирования, разрабатывать код программного продукта на основе готовой спецификации на уровне модуля, принципы построения баз данных, а также их администрирование.

К освоению программы допускаются лица, имеющие среднее профессиональное или высшее образование, в области информационных технологий.

Рекомендуется предварительное освоение таких дисциплин как «Основы алгоритмизации и программирования», «Информационные технологии».

МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

**«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»**

**(ГАПОУ ОКЭИ)**

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ:  Директор ГАПОУ «Оренбургский колледж экономики и информатики»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.А. Селиверстов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

# ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА

**ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

«Разработка программного обеспечения»

72 часа

Оренбург, 2020

**ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

1. **Цель программы**

Цель Программы – совершенствование у слушателей профессиональной компетенции, в области программирования и создания ИТ-продуктов с применением платформы .NET и SQL Management Studio.

1. **Планируемые результаты обучения:**

2.1. Знание (осведомленность в областях)

2.1.1. основные принципы технологии структурного и объектно-ориентированного программирования;

* + 1. основные этапы разработки программного обеспечения.
    2. разрабатывать объекты баз данных в выбранной СУБД;
    3. работать в профессиональной среде разработки;

2.2. Умение (способность к деятельности)

2.2.1 формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;

2.2.2 создавать программу по разработанному алгоритму как отдельный модуль.

2.3 Навыки (использование конкретных инструментов)

2.3.1 разрабатывать базовые алгоритмические конструкции;

2.3.2 проводить алгоритмизацию, кодирование, отладку и тестирование программного средства;

2.3.3 решать простые, сложные и нестандартные задачи в математической области;

2.3.4 строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;

2.3.5 разрабатывать код программного продукта на основе готовой спецификации на уровне модуля.

**3. Категория слушателей**

Образование:

К освоению программы допускаются лица, имеющие среднее профессиональное или высшее образование в области информационных технологий.

Наличие опыта профессиональной деятельности: не требуется

Рекомендуется предварительное освоение иных дисциплин таких как:

– дисциплина «Основы алгоритмизации и программирования»;

– дисциплина «Информационные технологии».

**4. Учебный план программы «Разработка программного обеспечения»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль** | **Всего,**  **час** | **Виды учебных занятий** | | |
| **лекции** | **практические занятия** | **самостоятельная работа**  **слушателя** |
| 1 | Теория баз данных. | 18 | 6 | 12 | - |
| 2 | Базовый язык реализации .NET решений – С#. | 22 | 8 | 12 | 2 |
| 3 | Взаимодействие с источниками данных с помощью ADO.NET. | 14 | 4 | 10 | - |
| 4 | Тестирование программного обеспечения. | 6 | 2 | 4 | - |
| 5 | Управления проектами. | 6 | 2 | 4 | - |
| Итоговая аттестация | | 6 | Контрольная работа с выполнением выбранного кейса | | |

**5. Календарный план-график реализации образовательной программы**

Периодичность набора групп (не менее 1 группы в месяц)

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование учебных модулей** | **Трудоемкость, (час)** | **Сроки обучения** |
| 1 | Теория баз данных. | 18 | 01.11.2020-05.11.2020 |
| 2 | Базовый язык реализации .NET решений – С#. | 22 | 06.11.2020-11.11.2020 |
| 3 | Взаимодействие с источниками данных с помощью ADO.NET. | 14 | 12.11.2020-16.11.2020 |
| 4 | Тестирование программного обеспечения. | 6 | 17.11.2020-18.11.2020 |
| 5 | Управления проектами. | 6 | 19.11.2020- 20.11.2020 |
| 6 | Итоговая аттестация | 6 | 21.11.2020 |
| **Всего:** | | 72 |  |

**6. Учебно-тематический план программы «Настройка, обновление и конфигурация операционных систем»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль/Тема** | **Всего, час** | **Виды учебных занятий** | | | **Формы контроля** |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | **Теория баз данных** | **18** | **6** | **12** | **-** | **-** |
| 1.1 | Введение в теорию баз данных, создание таблиц. Работа с данными insert, update, delete, select. | 4 | 2 | 2 |  | Тестирование |
| 1.2 | Агрегирующие функции, сортировка данных, приведение типов. | 2 |  | 2 |  |  |
| 1.3 | Многотабличные БД, связывание таблиц. | 2 |  | 2 |  |  |
| 1.4 | Понятие транзакций, изоляция транзакций. Понятие индекса | 2 |  | 2 |  |  |
| 1.5 | Группировка GROUP BY, HAVING | 2 |  | 2 |  |  |
| 1.6 | Объединение, понятие view, поворот таблиц. | 2 | 2 |  |  | Тестирование |
| 1.7 | Безопастность и резервное копирование БД. | 4 | 2 | 2 |  | Тестирование |
| 2 | **Базовый язык реализации .NET решений – С#.** | **22** | **8** | **12** | **2** |  |
| 2.1 | Введение в платформу Microsoft .NET. основы языка программирования C#, массивы и строки. | 4 | 2 | 2 |  | Письменный опрос |
| 2.2 | Введение в классы. Обработка исключений | 2 |  | 2 |  |  |
| 2.3 | Пространство имен. Перегрузка операторов. Индексаторы и свойства. | 2 |  |  | 2 |  |
| 2.4 | Наследование, интерфейсы. Структуры, перечисления | 2 |  | 2 |  |  |
| 2.5 | Делегаты, события | 2 |  | 2 |  |  |
| 2.6 | Введение в Generics. Сборка мусора | 2 | 2 |  |  | Тестирование |
| 2.7 | Взаиможействие с файловой системой. | 2 | 2 |  |  | Тестирование |
| 2.8 | Основы XML | 4 |  | 4 |  |  |
| 2.9 | Введение в LINQ | 2 | 2 |  |  |  |
| 3 | **Взаимодействие с источниками данных с помощью ADO.NET.** | **14** | **4** | **10** |  |  |
| 3.1 | Введение в ADO.NET. Присоединенный режим. | 6 | 2 | 4 |  | Письменный опрос |
| 3.2 | Фабрика провайдеров, ассинхронный режим доступа, конфигурационные файлы. Отсоединенный режим. | 4 | 2 | 2 |  | Тестирование |
| 3.3 | LINQ to SQL. Введение в EntityFramework. | 4 |  | 4 |  |  |
| 4 | **Тестирование программного обеспечения** | **6** | **2** | **4** |  |  |
| 4.1 | Введение в тестирование. Типичные программные ошибки и типы тестирования. | 2 | 2 |  |  | Письменный опрос |
| 4.2 | Утилиты для процесса тестирования. Автоматическое тестирование. | 2 |  | 2 |  |  |
| 4.3 | Unit-testing | 2 |  | 2 |  |  |
| 5 | **Управление проектами** | **6** | **2** | **4** |  |  |
| 5.1 | Введение в управление программными проектами. | 2 | 2 |  |  | Письменный опрос |
| 5.2 | Работа с Git. | 4 |  | 4 |  |  |
| 10 | Итоговая аттестация | **6** |  |  |  | Контрольная работа с выполнением кейса |

**7. Учебная (рабочая) программа повышения квалификации «Разработка программного обеспечения»**

**Раздел 1 Теория баз данных (18 часов).**

**Тема 1.1 Введение в теорию баз данных, создание таблиц. Работа с данными insert, update, delete, select (4 часа)**

**Содержание темы:** Понятия баз данных и назначение баз данных. Основные понятия реляционной модели. Язык запросов SQL. Предметная область. Создание таблиц баз данных. Типы данных. Основные операции с таблицами.

**Тема 1.2 Агрегирующие функции, сортировка данных, приведение типов (2 часа)**

**Содержание темы:** Агрегатные выражения. Вызовы оконных функций. Приведение типов CREATE CAST.

**Тема 1.3 Многотабличные БД, связывание таблиц (2 часа)**

**Содержание темы:** Понятия многотабличные базы данных. Оптимизация структуры данных. Поиск информации в базе данных. Схема данных. Многотабличные запросы.

**Тема 1.4 Понятие транзакций, изоляция транзакций. Понятие индекса (2 часа)**

**Содержание темы:** Уровни изоляции транзакции. Пример использования транзакций. Блокировки. Индексы по нескольким столбцам. Уникальные индексы. Индексы на основе выражений.

**Тема 1.5 Группировка GROUP BY, HAVING (2 часа)**

**Содержание темы:** Группировка, фильтрация групп.

**Тема 1.6 Объединение, понятие view, поворот таблиц (2 часа)**

**Содержание темы:** Что такое представление. Команда представления. Модифицирование представлений. Именование столбцов. Комбинирование предикатов представлений.

**Тема 1.7 Безопасность и резервное копирование БД (2 часа)**

**Содержание темы:** Основные определения. Принципы резервного копирования и восстановления данных. Стратегии резервного копирования. Выбор модели восстановления данных.

**Раздел 2 Базовый язык реализации .NET решений – С# (22 часа)**

**Тема 2.1 Введение в платформу Microsoft .NET. Основы языка программирования C#, массивы и строки (4 часа)**

**Содержание темы:** Роль платформы .NET. .NET Framework. Управляемый и не управляемый код. Создание проекта, изучение структуры окна платформы. Структура программы. Переменные, литералы, типы данных. Массивы и строки.

**Тема 2.2 Введение в классы. Обработка исключений (2 часа)**

**Содержание темы:** Создание классов, поиск блока catch при обработке исключений. Генерация исключений

**Тема 2.3 Пространство имен. Перегрузка операторов. Индексаторы и свойства (2 часа)**

**Содержание темы:** Индексаторы, применение нескольких параметров. Блоки get и set. Перегрузка индексаторов.

**Тема 2.4 Наследование, интерфейсы. Структуры, перечисления (2 часа)**

**Содержание темы:** Понятия наследования. Доступ к членам базового класса из класса-наследника. Конструкторы в приведенных классах. Порядок вызова конструкторов. Определения интерфейса. Модификаторы доступа интерфейсов. Применение интерфейса

**Тема 2.5 Делегаты, события (2 часа)**

**Содержание темы:** Определение делегатов. Применение делегатов. Анонимные методы работы с делегатами. Определение и вызов событий. Добавление обработчика событий. Добавление, удаление обработчиков. Управление обработчиками. Ковариантность и контравариантность делегатов

**Тема 2.6 Введение в Generics. Сборка мусора (2 часа)**

**Содержание темы:** Сборка мусора в C#. Класс System.GC

**Тема 2.7 Взаимодействие с файловой системой (2 часа)**

**Содержание темы:** Работа с дисками, каталогами, файлами. Классы file и FileInfo. Управление файлами. Чтение и запись файлов. Произвольный доступ к файлам.

**Тема 2.8 Основы XML (4 часа)**

**Содержание темы:** Что такое XML. Как применяется. Правила синтаксиса. Атрибуты, пространства имен, просмотр.

**Тема 2.9 Введение в LINQ (2 часа)**

**Содержание темы:** Три составляющих операций запроса. Источники данных. Выполнение запроса. Отложенное выполнение. Принудительное выполнение.

**Раздел 3 Взаимодействие с источниками данных с помощью ADO.NET (14 часов)**

**Тема 3.1 Введение в ADO.NET. Присоединенный режим (6 часов)**

**Содержание темы:** Введение в ADO.NET. Создание БД С# и SQL. Строка подключения. Создание подключения. Пул подключений.

**Тема 3.2 Фабрика провайдеров, асинхронный режим доступа, конфигурационные файлы. Отсоединенный режим (4 часа)**

**Содержание темы:** Общие сведения о модели фабрики. Фабричный конструктивный шаблон. Получение классов. Изменение данных. Отсоединённые режим.

**Тема 3.3 LINQ to SQL. Введение в EntityFramework (4 часа)**

**Содержание темы:** LINQ to DataSet. LINQ to SQL. LINQ to Entities.

**Раздел 4 Тестирование программного обеспечения (6 часов)**

**Тема 4.1 Введение в тестирование. Типичные программные ошибки и типы тестирования (2 часа)**

**Содержание темы:** Модели разработки ПО. Жизненный цикл тестирования. Требования к тестированию. Типичные ошибки при анализе требований.

**Тема 4.2 Утилиты для процесса тестирования. Автоматическое тестирование (2 часа)**

**Содержание темы**: Selenium. Katalon Studio. UFT. TestComplete.

**Тема 4.3 Unit-testing (2 часа)**

**Содержание темы:** Создание модульных тестов. Создание заглушек. Создание вручную. Написание тестов. Настройка времени ожидания для модульных тестов.

**Раздел 5 Управление проектами (6 часов)**

**Тема 5.1 Введение в управление программными проектами (2 часа)**

**Содержание темы:** Основные понятия и определение. Управление программными проектами. Комплексы стандартов управления.

**Тема 5.2 Работа с Git (4 часа)**

**Содержание темы:** Система контроля версий, установка, настройка, регистрация. Работа с системой контроля версий через консоль.

**Описание практико-ориентированных заданий и кейсов**

|  |  |  |
| --- | --- | --- |
| **Номер темы/модуля** | **Наименование практического занятия** | **Описание** |
| 1.1 | Создание таблиц базы данных. Работа с данными insert, update, delete, select. | Создайте таблицы базы данных по заданной диаграмме. Прописать запросы на редактирования баз данных. |
| 1.2 | Агрегирующие функции, сортирока данных, приведение типов. | Провести сортировку данных в базе данных. |
| 1.3 | Определение отношений между сущьностями. | Определить отношения между сущностями баз данных и основные ограничения. |
| 1.4 | Проведение транзакций в Базе данных. | Выполнить операции транзакции в базе данных. |
| 1.5 | Группировка GROUP BY, HAVING | Написать запросы группировки и фильтрации данных в базе. |
| 1.7 | Безопастность и резервное копирование БД. | Произвести резервное копирование и восстановление данных базы данных. |
| 2.1 | Решение практических задач с примененим массивов и преобразованием типов данных. | Простое число – это любое целое число, которое точно делится без остатка только само на себя и 1. Решето Эратосфена работает следующим образом:  1) Создайте массив, все элементы которого имеют начальное значение 1 (истина). Элементы массива с простыми индексами останутся равными 1. Все другие элементы массива, в конечном счете, установятся равными нулю.  2) Начиная с индекса массива 2 (индекс 1 должен быть простым) каждый раз отыскивается элемент массива с единичным значением, циклически обрабатывается оставшаяся часть массива и устанавливается в нуль каждый элемент массива, чей индекс кратен индексу элемента с единичным значением. Для индекса 2 все элементы в массиве с большим чем 2 индексом и кратные 2 установятся равными нулю (индексы 4, 6, 8, …); для индекса 3 все элементы с индексом свыше 3 и кратные 3, установятся равными нулю (индексы 6, 9, 12, …) и т.д.  3) Когда процесс закончится, элементы массива с единичным значением указывают, что их индексы – простые числа. Эти индексы могут быть напечатаны.  Напишите программу, которая использует массив из 100 млн элементов для определения и печати простых чисел между N\_min и N\_max. Элемент 0 массива во внимание не принимайте.  Сравните эффективность алгоритма с алгоритмом решения задачи «Простые числа». Критерий эффективности – время решения задачи. |
| 2.2 | Обработка исключительных ситуаций | Решение задач с использованием try …catch … finally |
| 2.4 | Блоки get и set. Перегрузка индексаторов. | Применить индексаторы для хранения двумерного массива или матрицы. |
| 2.5 | Доступ к членам базового класса из класса-наследника. Модификаторы доступа интерфейсов. Применение интерфейса | Настройки доступа. Наследование данных. |
| 2.8 | Язак разметки XML. Структура и обращение. | Определить структуру окна с помощью разметки XML. |
| 3.1 | Создание БД С# и SQL. Строка подключения. Создание подключения. Пул подключений. | Создать подключение к серверу с помощью компонентов ADO.NET |
| 3.2 | Фабричный конструктивный шаблон. Получение классов. Изменение данных. | Реализовать паттерн проектирования. Фабричный метод. |
| 3.3 | LINQ to DataSet. LINQ to SQL. LINQ to Entities. | Составить запросы используя сборки бизнес-логики |
| 4.2 | Реализация тестоваго сценария через Selenium. | Реализация базового класса для всех тестов и JUnit RuleChain |
| 4.3 | Создание модульных тестов. Создание заглушек. | Слой доступа к данным (MS SQL Server) |
| 5.2 | Система контроля версий, установка, настройка, регистрация. Работа с системой контроля версий через консоль. | Разворачивание сервера Git, создание репозитория. Работа через консоль. |

**8.Оценочные материалы по образовательной программе**

**8.1. Вопросы тестирования по модулям**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ модуля** | **Вопросы входного тестирования** | **Вопросы промежуточного тестирования** | **Вопросы итогового тестирования** |
| 1.1 | Что такое базы данных и зачем они нужны. Понятия реляционных модели данных.  Язык запросов SQL. | Какие группы операторов выделяются в составе языка SQL?  Дайте неформальное определение основных понятий реляционной модели  данных: отношение, кортеж, атрибут.  Для чего нужны внешние ключи в реляционных таблицах?  Что такое потенциальный ключ?  Предложите пример избыточного потенциального ключа для одной из таблиц | В текущей реализации базы данных «Авиаперевозки» предполагается, что самолеты одной модели могут иметь только одну компоновку салона. Представим, что руководством принято решение о том, что нужно учитывать возможность наличия различных компоновок для каждой модели? Какие таблицы придется модифицировать в таком случае и каким образом? Потребуется ли создавать дополнительные таблицы? |
| 1.7 | Выгрузка данных. | Как произвести холодное сохранение файлов в БД?  Как произвести горячее сохранение файлов? | Как произвести возврат режима резервного копирования?  Как производится инкрементальное резервное копирование? |
| 2.1 | Роль платформы .NET. .NET Framework. | Как создать управляемый код?  Как создать не управляемый код?  Как создать проект? изучение структуры окна платформы. | Структура программы?  Что такое переменные?  Что такое литералы?  Какие типы данных есть в C#?  Что такое массивы?  Как выполнить преобразования в строку? |
| 2.6 | Сборка мусора в C# | Что такое ссылочные типы?  Для чего используется класс System.GC?  Какие методы существуют у класса System.GC? | Как выполнить сборку мусора до объектов первого поколения?  Как создать диструкторы? |
| 2.7 | Взаиможействие с файловой системой. | С какой файловой системой начинается работа с файловыми службами?  Какие классы предназначены для работы с пространством имен?  Как получить список файлов и подкаталогов? | Как получить информацию о каталоге?  Как переместить каталог?  Какими методами обладает класс FileInfo? |
| 2.9 | Введение в LINQ | Три составляющие операции запроса?  Источник данных?  Запрос?  Выполнение запроса? | Как получить источник данных?  Как создать запрос?  Как выполнить запрос? |
| 3.1 | Технология ADO.NET | Что такое ADO.NET?  Какие провайдеры имеет ADO.NET?  Какие пространства имен использует ADO.NET? | Как создать подключение по технологии доступа к данным? |
| 3.2 | Фабрика провайдеров, ассинхронный режим доступа, конфигурационные файлы. | Что такое паттерн проектирования?  Что такое порождающие паттерны?  Что такое структурные паттерны? | В чем заключается фабричный метод?  Как работает логика фабричного метода?  Как реализовать фабричный метод? |
| 4.1 | Введение в тестирование. Типичные программные ошибки и типы тестирования. | Что такое тестирование?  Модели разработки программного обеспечения?  Какие методы тестирования существуют? | Что такое требование?  Типичные ошибки при выявлении требований к тестированию? |
| 5.1 | Введение в управление программными проектами. | Кем осуществляется инициализация проекта?  Какие этапы должно пройти планирование проекта? | Как идентифицировать риски от проектов?  Главные риски программных проектов? |

**8.2. Описание показателей и критериев оценивания, шкалы оценивания**

При усвоении каждой темы за текущую учебную деятельность слушателя выставляются оценки по 4-балльной шкале. Оценка за каждое задание в процессе текущей учебной деятельности определяется на основе процентного отношения операций, правильно выполненных слушателем во время выполнения задания:

90-100 % – «5»,

75-89 % – «4»,

60-74 % – «3»,

менее 60 % – «2».

**8.3. Примеры контрольных заданий по модулям или всей образовательной программе**

**Контрольное задание по всей образовательной программе**

|  |  |  |
| --- | --- | --- |
| № | Задание | Максимальный вес выполненного задания |
| 1 | На основании выбранного варианта выполнить следующее:  1. Выполнить анализ предметной области исследуемой организации;  2. Описать основные сущности предметной области;  3. Расставить существующие связи между сущностями: самостоятельно  добавить в каждую сущность первичные ключи и установить внешние  ключи между сущностями;  4. Построить инфологическую модель базы данных организации;  5. Построить даталогическую модель базы данных организации.  БД – торговая организация. БД состоит из следующих таблиц: торговая  организация, торговая точка, продавцы, поставщики, заказы  поставщикам.  Таблица торговая организация имеет следующие атрибуты: название  торговой организации, адрес, ФИО директора, налоговый номер.  Таблица торговая точка имеет следующие атрибуты: название торговой  точки, тип (универмаги, магазины, киоски, лотки и т.д.), торговая  организация, адрес, ФИО заведующего.  Таблица продавцы имеет следующие атрибуты: фамилия, имя, отчество,  торговая точка, должность, год рождения, пол, адрес проживания, город.  Таблица поставщики имеет следующие атрибуты: название поставщика,  тип деятельности, страна, город, адрес.  Таблица заказы поставщикам имеет следующие атрибут | 15% |
| 2 | 1. Создать на сервере pi\_srv (или на локальном компьютере, если нет сервера) рабочую папку для хранения файлов, получаемых при выполнении практической работы. Эта папка должна располагаться в папке \Базы данных\Группа\Студент и соответствовать номеру выполняемой практической работы. 2. На основании индивидуального задания выбрать имя файла создаваемой базы данных. Для имени лучше всего выбрать одно или несколько английских слов, соответствующих наименованию предметной области. Использование для имени русских слов, записанных латинскими буквами, не допускается. 3. Открыть приложение " Среда SQL Server Management Studio ". Для этого можно либо воспользоваться меню Пуск (Пуск/Программы/ Microsoft SQL Server 2008 / Среда SQL Server Management Studio). 4. Создать соединение с локальным или удаленным сервером. 5. Создать базу данных для своей предметной области с помощью диалога, выбрав сервер "pi\_srv" или локальный сервер “Имя\_компьютера\SQLEXPRESS” 6. Создать базу данных и указать в качестве имени файла "\Базы данных\Группа\ФИО\_студента\Название\_БД". 7. Извлечь метаданные для автоматической генерации команды создания базы данных. 8. Удалить базу данных, выполнив команду "Database/Drop Database" (База данных/Удалить базу данных). 9. Создать базу данных вторым способом, выполнив в окне "Script Executive" операторы, полученные при извлечении метаданных перед предыдущим удалением. 10. Создать резервную копию базы данных. 11. Удалить базу данных. 12. Восстановить базу данных из резервной копии. 13. Сохранить файл сценария на сервере в папке "Студент", дав ему имя «лаб.№1» и стандартное расширение "\*.sql". | 30% |
| 3 | На основании выбранного варианта выполнить следующее: 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – успеваемость студентов ВУЗА. БД состоит из следующих таблиц: факультеты, кафедры, учебные группы, студенты, ведомости успеваемости. 27 Таблица факультеты имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон. Таблица кафедра имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей. Таблица учебные группы имеет следующие атрибуты: название группы, год поступления, курс обучения, кол-во студентов в группе. Таблица студенты имеет следующие атрибуты: студента, фамилия, имя, отчество, группа, год рождения, пол, адрес, город, телефон. Таблица ведомости успеваемости имеет следующие атрибуты: группа, студент, предмет, оценка. | 15% |
| 4 | Создайте форму для работы со справочником Courses. Форма должна позволять вносить новые записи, удалять имеющиеся, вносить поправки. Отображение данных таблицы Courses выполните с помощью любых элементов управления Windows-форм, кроме элемента DataGrid. Для отображения в форме данных, хранящихся в формате Date, используйте элемент управления датой/временем DateTimePicker, предназначенный для отображения и редактирования даты/ | 30% |
| 5 | Скачать и установить систему Tortoise SVN, Git или TFS на выбор.  Создать локальный репозиторий  Выложить в репозиторий тестовый проект Visual Studio  Добавить в тестовый проект новый класс. Изменить существующий код. Выложить в репозиторий  Осуществить откат к старой версии выложенного в репозиторий проекта  Удалить локальную копию проекта и скачать последнюю версию из репозитория | 10% |
| Итого максимальная оценка за выполнение | | 100% |

**8.4. Тесты и обучающие задачи (кейсы), иные практикоориентированные формы заданий**

|  |  |  |
| --- | --- | --- |
| **№** | **Форма задания** | **Формулировка** |
| 1 | Кейс | Создать приложение HumanResources, включающее базу данных под управлением SQL Server и Windows-формы, как интерфейс пользователя к базе данных.  Решение  Использовать Visual Studio.NET для создания базы данных HumanResources и элементы управления Windows-форм для разработки интерфейса подключения, отображения и редактирования данных из базы данных HumanResources.  Обсуждение  Рассмотрим создание базы данных, предназначенной для хранения следующей информации о каждом сотруднике компании:  название отдела, в котором работает служащий;  размер полученной сотрудником заработной платы;  размер иных доходов сотрудников;  перечень курсов, которые посещал служащий компании в целях повышения квалификации.  Для создания таблиц и отношений между ними можно использовать инструменты SQL Server, но также можно использовать и средства среды Visual Studio.NET. |
| 2 | Кейс | Создать на языке Transact-SQL файл базы данных согласно номеру варианта (присвоить ей новое имя, несовпадающие с именем базы данных созданной в лаб.№3).  База данных разрабатывается на основе спроектированной концептуальной модели данных в лаб.№1.  Создать программно на языке SQL все таблицы, с указанием первичных и внешних ключей и ограничения целостности. Все программные инструкции команд SQL сохранять в файлах с расширением \*.sql в папке ФИО\_студента/Лаб4.  Заполнить таблицы данными по 5 записей в каждой.  Создать текстовый отчет, в котором отобразить sql-команды разработанных запросов и скриншоты результатов работы из СУБД SQL Server Management Studio |

**8.5. Описание процедуры оценивания результатов обучения**

Процедура оценивания результатов обучения заключается в выполнении **контрольной итоговой работы и выполнения выбранного кейса.** Проверка кейса осуществляется в дистанционном автоматизированном формате с использованием оборудования образовательной организации.

**9. Организационно-педагогические условия реализации программы**

**9.1. Кадровое обеспечение программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№**  **п/п** | **Фамилия, имя, отчество (при наличии)** | **Место основной работы и должность, ученая степень и ученое звание (при наличии)** | **Ссылки на веб-страницы с портфолио (при наличии)** | **Фото в формате jpeg** | **Отметка о полученном согласии на обработку персональных данных** |
| 1 | Асташева Татьяна Сергеевна | государственное автономное профессиональное образовательное учреждение «Оренбургский колледж экономики и информатики», преподаватель |  | **D:\Работа\Учебный центр\Цифровые сертификаты\Асташева.jpg** |  |

**9.2. Учебно-методическое обеспечение и информационное сопровождение**

|  |  |
| --- | --- |
| **Учебно-методические материалы** | |
| **Методы, формы и технологии** | **Методические разработки,**  **материалы курса, учебная литература** |
| Проектный метод; технологии имитационного моделирования;  модели поискового обучения | Эндрю Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5. Вильямс, 2015. - 1312 c  Адам Фримен. ASP.NET MVC 5 с примерами на C# 5.0 для профессионалов. Вильямс, 2017. - 736 c  С. Р. Гуриков. Введение в программирование на языке Visual C#. ДРОФА, 2013. - 448 c  Роберт С. Мартин,Мика Мартин. Принципы, паттерны и методики гибкой разработки на языке C#. Символ-Плюс, 2011. - 768 c  Джозеф Албахари,Бен Албахари. C# 5.0. Карманный справочник. Вильямс, 2013. - 288 c  Эндрю Троелсен, Филипп Джепикс. Язык программирования C# 6.0 и платформа .NET 4.6. Вильямс, 2017. - 1440 c  Code–basics [Электронный ресурс], – Режим доступа: [Code-basics.ru](https://code-basics.ru/) – 15.10.2020  Федорова, Г. Н. Разработка и администрирование баз данных: учебник для студ. учреждений СПО. – М.: Академия, 2015.  GitHub платформа для разработки [Электронный ресурс], – Режим доступа: <https://github.com/> – 15.10.2020 |
| Проектный метод; технологии имитационного моделирования;  модели поискового обучения | Программа обеспечивается учебно-методическим материалами по всем учебным модулям, включающим:  текстовые и презентационные материалы по модулю; задания для аудиторной и самостоятельной работы; оценочные средства для итоговой аттестации; учебно-методические материалы для изучения. |

|  |  |
| --- | --- |
| **Информационное сопровождение** | |
|  | Электронные  информационные ресурсы |
|  | GitHub платформа для разработки [Электронный ресурс], – Режим доступа: <https://github.com/> – 15.10.2020 |
|  | Code–basics [Электронный ресурс], – Режим доступа: [Code-basics.ru](https://code-basics.ru/) – 15.10.2020 |

**9.3. Материально-технические условия реализации программы**

|  |  |
| --- | --- |
| Вид занятий | Наименование оборудования,  программного обеспечения |
| Лекция | АРМ в составе: системный блок (i5 9500 3.00 ГГц 9 МБ/В360/2\*8 ГБ 2666 МГц DDR4/250 ГБ M.2 SSD/2LAN/300 Вт), монитор Dell P2419H, клавиатура Logitech K120, мышь Logitech B100;  – лицензионное программное обеспечение: обеспечение: ОС Windows 10, SQL Server Management Studio 2017; Microsoft Visual Studio 2019; установленный сервер Git.. |
| Практическое занятие | – сервер Fujitsu PRIMERGY RX2540 M5 12x3.5 (2\*Xeon Silver 4210 2.20 ГГц/8\*16 ГБ 2Rx8 DDR4-2933 RECC/5\*480 ГБ SAS SSD RI/RAID EP540i/4\*1 Гб OCP LAN/2\*800 Вт/Rail/ARM/3y OS, 24x7, 4h);  – АРМ в составе: системный блок (i7 9500 3.00 ГГц 9 МБ/В360/2\*8 ГБ 2666 МГц DDR4/250 ГБ M.2 SSD/2LAN/300 Вт), монитор Dell P2419H, клавиатура Logitech K120, мышь Logitech B100;  – лицензионное программное обеспечение: ОС Windows 10, SQL Server Management Studio 2017; Microsoft Visual Studio 2019; установленный сервер Git. |

**III. Паспорт компетенций**

ПАСПОРТ КОМПЕТЕНЦИИ

Наименование дополнительной профессиональной образовательной программы повышения квалификации: **«Разработка программного обеспечения»**

Наименование организации, реализующей дополнительную профессиональную образовательную программу повышения квалификации: **государственное автономное профессиональное образовательное учреждение «Оренбургский колледж экономики и информатики»**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Наименование компетенции | Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. | |
| 2 | Указание типа компетенции | Профессиональная | |
| 3 | Определение, содержание и основные сущностные характеристики компетенции | Под компетенцией понимается умение оптимально реализовывать алгоритмы построения на заданном языке программирования, а также применять меры к безопасности хранения данных. | |
| 4 | Дескриптор знаний, умений и навыков по уровням | Уровни сформированности компетенции  обучающегося | Индикаторы |
|  | Начальный уровень | Умеет:  разрабатывать алгоритмы для конкретных задач;  - использовать программы для графического отображения алгоритмов;  - реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;  - выполнять проверку, отладку кода программы;  Знать:  – понятие алгоритмизации, свойства алгоритмов, общие принципы построения  алгоритмов, основные алгоритмические конструкции;  - эволюцию языков программирования, их классификацию, понятие системы программирования;  - основные элементы языка, структуру программы, операторы и операции, управляющие  структуры, структуры данных, файлы, классы памяти;  Владеет навыками:  – разработки программного кода на уровне готовой спецификации;  – создание баз данных. |
|  | Базовый уровень | Умеет:  – разрабатывать алгоритмы для конкретных задач;  - использовать программы для графического отображения алгоритмов;  - определять сложность работы алгоритмов;  - работать в среде программирования;  - выполнять проверку, отладку кода программы;  Знать:  – понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;  - эволюцию языков программирования, их классификацию, понятие системы  программирования;  - основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;  - подпрограммы, составление библиотек подпрограмм;  -  Владеет навыками:  – разработки программного кода на уровне готовой спецификации;  – создание баз данных. |
|  | Продвинутый | Умеет:  – разрабатывать алгоритмы для конкретных задач;  - использовать программы для графического отображения алгоритмов;  - определять сложность работы алгоритмов;  - работать в среде программирования;  - реализовывать построенные алгоритмы в виде программ на конкретном языке  программирования;  Знать:  – понятие алгоритмизации, свойства алгоритмов, общие принципы построения  алгоритмов, основные алгоритмические конструкции;  - эволюцию языков программирования, их классификацию, понятие системы программирования;  - основные элементы языка, структуру программы, операторы и операции, управляющие  структуры, структуры данных, файлы, классы памяти;  - подпрограммы, составление библиотек подпрограмм;  - объектно-ориентированную модель программирования, основные принципы объектноориентированного программирования на примере алгоритмического языка;  Владеет навыками:  – разработки программного кода на уровне готовой спецификации;  – разработки пользовательских сценариев тестирования;  – работа с командной строкой системы контроля версий;  – создание баз данных. |
|  | Профессиональный | Умеет:  – разрабатывать алгоритмы для конкретных задач;  - использовать программы для графического отображения алгоритмов;  - определять сложность работы алгоритмов;  - работать в среде программирования;  - реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;  - оформлять код программы в соответствии со стандартом кодирования;  - выполнять проверку, отладку кода программы;  Знать:  понятие алгоритмизации, свойства алгоритмов, общие принципы построения  алгоритмов, основные алгоритмические конструкции;  - эволюцию языков программирования, их классификацию, понятие системы программирования;  - основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;  - подпрограммы, составление библиотек подпрограмм;  - объектно-ориентированную модель программирования, основные принципы объектноориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.  Владеет навыками:  – разработки программного кода на уровне готовой спецификации;  – разработки пользовательских сценариев тестирования;  – работа с командной строкой системы контроля версий;  – создание баз данных. |
| 5. | Характеристика взаимосвязи данной компетенции с другими компетенциями/ необходимость владения другими компетенциями для формирования данной компетенции | Формируемые компетенции цифровой грамотности в рамках освоения курса включает в себя изучение паттернов проектирования баз данных. А также методов реализации программного кода приложения. Также используются технологии ADO.NET для возможности создания строк подключения к базам данных. Рассматривается развёртывание Git сервера, создание репозиториев для дальнейшей работы с ними. Методы тестирования и создания тест кейсов для проверки работоспособности программного обеспечения. | |
| 6. | Средства и технологии оценки | Контрольные вопросы, кейсы | |

**Возможные сценарии профессиональной траектории граждан** по итогам освоения образовательной программы (в соответствии с приложением)

|  |  |
| --- | --- |
| **Цели получения персонального цифрового сертификата** | |
| **Текущий статус** | **Цель** |
| **Трудоустройство** | |
| состоящий на учете в Центре занятости | трудоустроенный, самозанятый (фриланс) |
| Безработный |
| безработный по состоянию здоровья |
| **Развитие компетенций в текущей сфере занятости** | |
| работающий по найму в организации, на предприятии | сохранение текущего рабочего места |
| развитие профессиональных качеств |
| повышение заработной платы |
| смена работы без изменения сферы  профессиональной деятельности |
| временно отсутствующий на рабочем месте (декрет, отпуск по уходу за ребенком и др.) | сохранение и развитие квалификации |
| **Переход в новую сферу занятости** | |
| освоение новой сферы занятости | расширение кругозора |
| освоение смежных профессиональных областей | расширение профессиональной деятельности |