

Требования и рекомендации

Введение

Настоящий документ определяет правила выполнения экзаменационного задания для компетенции «Программные решения для бизнеса».

Для выполнения задач экзаменационного задания вы можете использовать любые инструменты, предоставляемые согласно инфраструктурного листа.

В случае нехватки времени для выполнения всех оставшихся задач вы можете пропускать выполнение некоторых задач в пользу других. Однако ожидается, что вы предоставите максимально завершенную работу в конце каждой сессии, чтобы облегчить оценку вашей работы.

Правила

Во время проведения экзамена необходимо соблюдать следующие правила:

- Запрещен доступ в Интернет (кроме разового доступа в течение сессии не более 15 минут);
- Запрещено использование любых гаджетов (мобильный телефон, планшет, смарт-часы и т.д.);
- Запрещено использование ваших собственных устройств хранения данных (USB-накопители, жесткие диски и т.д.);
- Запрещено общение с другими участниками экзамена;
- Запрещено приносить на экзамен книги, заметки и т.д.;
- Разрешено использовать личные устройства ввода информации (клавиатура, мышь, трекбол и т.д.), но эти устройства должны быть проводными, непрограммируемыми и должны работать без дополнительной установки драйверов (эти требования предварительно проверяются техническим экспертом);
- Разрешено использовать личные средства повышения эргономики (коврик для мыши, подставка под запястья и т.д.), а также талисманы (также проходят проверку у технического эксперта);
- При возникновении любой внештатной ситуации с программным или аппаратным обеспечением, а также периферийными устройствами необходимо немедленно прервать работу и обратиться к эксперту.

Несоблюдение этих правил может привести к удалению с площадки проведения экзамена.

Название приложения

Используйте соответствующие названия для ваших приложений и файлов. Так, например, наименование настольного приложения должно обязательно включать название компании-заказчика.

Файловая структура

Файловая структура проекта должна отражать логику, заложенную в приложение. Например, все формы содержатся в одной директории, пользовательские визуальные компоненты – в другой, классы сущностей – в третьей.

Структура проекта

Каждая сущность должна быть представлена в программе как минимум одним отдельным классом. Классы должны быть небольшими, понятными и выполнять одну единственную функцию (Single responsibility principle).

Для работы с разными сущностями используйте разные формы, где это уместно.

Логическая структура

Логика представления (работа с пользовательским вводом/выводом, формы, обработка событий) не должна быть перемешана с бизнес-логикой (ограничения и требования, сформулированные в заданиях), а также не должна быть перемешана с логикой доступа к базе данных (SQL-запросы, запись, получение данных). В идеале это должны быть три независимых модуля.

Руководство по стилю

Визуальные компоненты должны соответствовать руководству по стилю, предоставленному в качестве ресурсов к заданию в соответствующем файле. Обеспечьте соблюдение требований всех компонентов в следующих областях:

- цветовая схема,
- размещение логотипа,
- использование шрифтов,
- установка иконки приложения.

Макет и технические характеристики

Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

- разметка и дизайн (предпочтение отдается масштабируемой компоновке);
- должно присутствовать ограничение на минимальный размер окна;
- должна присутствовать возможность изменения размеров окна, где это необходимо;
- увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);
- группировка элементов (в логические категории);
- использование соответствующих элементов управления (например, выпадающих списков для отображения подстановочных значений из базы данных);
- расположение и выравнивание элементов (метки, поля для ввода и т.д.);
- последовательный переход фокуса по элементам интерфейса (по нажатию клавиши TAB);
- общая компоновка логична, понятна и проста в использовании;
- последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например, с помощью кнопки «Назад»);
- соответствующий заголовок на каждом окне приложения (не должно быть значений по умолчанию типа MainWindow, Form1 и тп).

Обратная связь с пользователем

Уведомляйте пользователя о совершаемых им ошибках или о запрещенных в рамках

задания действиях, запрашивайте подтверждение перед удалением, предупреждайте о неотвратимых операциях, информируйте об отсутствии результатов поиска и т.п. Окна сообщений соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления. Также можно использовать визуальные подсказки для пользователя при вводе данных.

Обработка ошибок

Не позволяйте пользователю вводить некорректные значения в текстовые поля сущностей. Например, в случае несоответствия типа данных или размера поля введенному значению. Оповестите пользователя о совершенной им ошибке.

Обратите внимание на использование абсолютных и относительных путей к изображениям. Приложение должно корректно работать, в том числе и при перемещении папки с исполняемым файлом.

При возникновении непредвиденной ошибки приложение не должно аварийно завершать работу.

Оформление кода

Идентификаторы переменных, методов и классов должны отражать суть и/или цель их использования, в том числе и наименования элементов управления (например, не должно быть значений по умолчанию типа `Form1`, `button3`).

Идентификаторы должны соответствовать соглашению об именовании (Code Convention) и стилю `CamelCase` (для `C#` и `Java`) и `snake_case` (для `Python`).

Допустимо использование не более одной команды в строке.

Комментарии

Используйте комментарии для пояснения неочевидных фрагментов кода. Запрещено комментирование кода.

Хороший код воспринимается как обычный текст. Не используйте комментарии для пояснения очевидных действий. Комментарии должны присутствовать только в местах, которые требуют дополнительного пояснения.

Используйте тип комментариев, который в дальнейшем позволит сгенерировать XML- документацию, с соответствующими тегами (например, `param`, `return(s)`, `summary` и др.)

Оценка

Каждая задача оценивается путем тестирования реализации требуемого функционала. Так как требования к реализуемой системе очень высоки, возможно, будут использоваться средства для автоматизированного тестирования приложения. В связи с этим, в ходе разработки, может возникнуть необходимость следовать определенным правилам именования и структурирования проекта.

Предоставление результатов

Все практические результаты должны быть переданы заказчику путем загрузки файлов на предоставленный вам репозиторий системы контроля версий `git`. Практическими результатами являются:

- исходный код приложения (в виде коммита текущей версии проекта, но не архивом),
- исполняемые файлы,
- прочие графические/текстовые файлы.

Результаты работы каждой сессии должны быть загружены в отдельный репозиторий с названием «Сессия X» (X – номер сессии).

Для оценки работы будет учитываться только содержимое репозитория. При оценке рассматриваются заметки только в электронном виде (readme.md). Рукописные примечания не будут использоваться для оценки.

Проект обязательно должен содержать описание в формате Markdown (см. шаблон в файле README-Template.md или README-Template_rus.md). Заполните также дополнительную информацию о проекте и способе запуска приложения в файле readme.md.

Обратите внимание, что дополнительного времени после окончания сессии на сохранение не предусмотрено, поэтому будьте бдительны и загружайте результаты работ своевременно в рамках сессии.

Описание предметной области

Вашей задачей является разработка информационной системы для точки проката горнолыжного курорта «Игора» в Ленинградской области.

Курорт «Игора» в Ленинградской области — место для отдыха в любое время года. В зимнее время здесь функционирует горнолыжный центр, летом — «Озеро развлечений» с пляжной зоной и спортивными площадками.

Открытие горнолыжного курорта «Игора» состоялось в январе 2006 года. К этой дате возвели трассы с подъемниками, гостиницу у подножия горы. Через год на курорте был построен спа-центр с бассейном и ресторан «Ле Шале». В 2010 году был введен в эксплуатацию центр зимних видов спорта «Ледовый дворец».

В настоящее время, «Игора» — один из самых популярных курортов Ленинградской области. Он неоднократно становился местом проведения соревнований по зимним видам спорта — как российского, так и международного уровня.

На горнолыжном курорте «Игора» оборудовано 10 трасс для катания разного уровня. Склоны имеют протяженность от 530 до 1200 метров. Среди них 2 склона для обучения катанию на горных лыжах и сноуборде, один склон для катания на ватрушках. Перепад высот на трассах «Игоры» составляет от 88 до 116 метров.

Общая протяженность трасс — 4,2 км.

На курорте «Игора» работает точка проката оборудования. Клиенты могут взять в аренду оборудование, обувь, защитные материалы и др.

Требования к ИС:

- работа с единой базой данных для всех пользователей;
- ведение базы заказов;
- составление отчетов;
- асинхронное выполнение всех операций (приложение не должно «висеть»).

Используемые технологии при разработке:

- разработка клиент-серверных приложений;
- создание штрих-кодов.

Порядок работы точки проката:

На точке проката работают администратор, старший смены и продавец. Продавец может сформировать заказ на аренду, старший смены может сформировать заказ и принять товар; администратор точки проката может сформировать отчеты, проконтролировать всех пользователей по истории входа, работать с данными о расходных материалах, используемых на точке проката.

Прием и оформление заказа выполняется продавцом или старшим смены.

Продавец или старший смены оформляют заказ на прокат, на каждый заказ формируется штрих-код, распечатывается и наклеивается на оборудование для проката. Если клиент берет несколько позиций, то штрих-код печатается в нескольких экземплярах и наклеивается на каждое оборудование. Принимает оборудование старший смены на складе после проката, приемка происходит путем считывания по штрих-коду. При считывании штрих-кода в базе данных фиксируется возврат оборудования и закрываются заказы.

Администратор точки проката может отслеживать все заказы и составлять отчеты по арендуемым позициям для дальнейшего анализа.

Руководство по стилю

Общие требования

При создании приложения руководствуйтесь требованиями, описанными в документе «Требования и рекомендации.pdf». Не допускайте орфографические и грамматические ошибки.

Использование логотипа

Все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом (в ресурсах). Не искажайте логотип (не изменяйте изображение, его пропорции, цвет).

Также для приложений должна быть установлена иконка.

Шрифт

Используйте шрифт Comic Sans MS.

Цветовая схема

В качестве основного фона используется белый цвет; в качестве дополнительного: RGB (118, 227, 131).

Для акцентирования внимания пользователя на целевое действие интерфейса используйте цвет RGB (73, 140,81).

Основной фон	Дополнительный фон	Акцентирование внимания
RGB (255, 255, 255)	RGB (118, 227, 131)	RGB (73, 140,81)

Сессия 1

Модуль 1: Анализ и проектирование требований, бизнес-процессов

Диаграмма прецедентов

Для согласования процесса разработки с заказчиком горнолыжного курорта «Игора» Вам необходимо ознакомиться с описанием предметной области и заданием экзамена, сделать диаграмму прецедентов (Use Case) для основных пользователей системы. Сохраните файл с диаграммой в форматах .vsdx и .pdf.

Модуль 2: Проектирование реляционного хранилища данных

Проектирование базы данных (ERD)

Изучите описание предметной области.

На основе описания предметной области и задания демонстрационного экзамена (все сессии) Вам необходимо спроектировать ER-диаграмму для информационной системы точки проката горнолыжного курорта «Игора». Обязательна 3 нормальная форма с обеспечением ссылочной целостности. При разработке диаграммы обратите внимание на согласованную осмысленную схему именования, создайте необходимые первичные и внешние ключи, определите ограничения внешних ключей, отражающие характер предметной области.

ER - диаграмма должна быть представлена в формате .pdf и .vsdx и содержать таблицы, связи между ними, атрибуты и ключи (типами данных на данном этапе можно пренебречь).

Проверка ER-диаграммы будет выполнена по итогам работы двух сессий.

Модуль 3: Разработка баз данных, объектов баз данных и импорт

База данных

Для работы приложения вам необходимо создать базу данных точки проката горнолыжного курорта «Игора». Создайте базу данных, используя предпочтительную платформу, на сервере баз данных, который вам предоставлен.

Создайте таблицы основных сущностей, атрибуты, отношения и необходимые ограничения. В любом случае созданные таблицы должны содержать начальные тестовые данные.

При получении нового задания вам необходимо будет создать новые объекты в уже существующую базу данных.

Импорт данных

Заказчик системы предоставил файлы с данными (с пометкой import в ресурсах) для переноса в новую систему. Подготовьте данные файлов для импорта и загрузите в разработанную базу данных.

Data Dictionary

Для диаграммы ER необходимо создать словарь данных – набор информации, описывающий, какой тип данных хранится в базе данных, их формат, структуру и способы использования данных. Обратите внимание на соответствие вашей диаграммы и словаря данных. Используйте подходящие типы данных, ограничения и форматы. Отрадите в документе ограничения, определенные в таблицах, включая первичные ключи, отношения внешнего ключа с другими таблицами и ненулевые ограничения. Не забудьте сделать необходимые пояснения и комментарии к неоднозначным полям. В качестве шаблона словаря необходимо использовать файл DataDictionary_Template.xlsx. Сохраните итоговый документ со словарем данных в формате .xlsx.

Проверка словаря данных будет выполнена по итогам работы двух сессий.

Модуль 4: Разработка desktop-приложений

Окно входа для сотрудников точки проката горнолыжного курорта «Игора»

При запуске приложения окно входа – первое, что видит пользователь. На ней пользователю предлагается ввести свой логин и пароль. Только после удачной авторизации пользователь получает доступ к остальным модулям системы точки проката горнолыжного курорта «Игора».

При вводе пароль должен быть скрыт маской ввода, но также должна быть реализована возможность просмотра введенного пароля.

При входе система выводит фото пользователя, фамилию и имя пользователя, его роль.

После авторизации пользователь получает доступ к нужному функционалу:

- продавец может сформировать заказ;
- старший смены может сформировать заказ и принять товар;
- администратор точки проката может сформировать отчеты, проконтролировать всех пользователей по истории входа, работать с данными о расходных материалах, используемых на точке проката.

Реализуйте необходимые интерфейсы для всех пользователей системы. После входа в любую учетную запись должна быть реализована возможность выхода на главный экран – окно входа.

При входе в учетную запись должен быть виден таймер (часы:минуты), который фиксирует время сеанса пользователя. Сеанс пользователя не должен превышать 2 ч 30 минут, так как через каждые 2 ч 30 минут необходимо выполнить проверку кассы и внести запись в кассовую книгу. За 15 минут до окончания времени сеанса должно появиться сообщение об окончании времени сеанса. По окончании времени сеанса реализуйте выход из учетной записи и блокировку входа на 15 минут.

Для удобства проверки экспертной группой - укажите время сеанса – 10 минут, появление сообщения – за 5 минут до окончания времени сеанса, блокировка входа – 3 минуты.

После второй попытки неуспешной авторизации система выдает сообщение о неуспешной авторизации, а затем помимо ввода логина и пароля просит ввести captcha, состоящую из 3 символов (цифры и буквы латинского алфавита) и графического шума.

САРТСНА - должна содержать минимум 3 символа (буква или цифра), которые выведены не в одной линии. Символы должны быть либо перечеркнуты, либо наложены друг на друга.

Реализуйте возможность повторной генерации captcha, если пользователю непонятны символы из-за шума.

После попытки неудачной авторизации с вводом captcha, система блокирует возможность входа на 10 секунд.

История входа

Приложение должно хранить историю входа в систему, так как в системе будут храниться данные по всем заказам точки проката горнолыжного курорта «Игора». Окно для просмотра истории должно быть доступно администратору точки проката. В этом окне необходимо реализовать просмотр всей истории входа, а также фильтрацию по логину пользователя. Кроме этого, необходимо добавить сортировку по дате попытки входа. Каждая запись истории должна содержать следующие данные: время, логин пользователя, успешная или ошибочная попытка входа.

Оформление заказа

Прием и оформление заказа выполняется продавцом или старшим смены точки проката горнолыжного курорта «Игора».

Продавец или старший смены оформляют заказ на прокат, на каждый заказ формируется штрих-код, распечатывается и наклеивается на оборудование для проката. Прием оборудования на складе после проката происходит только по штрих-коду.

Реализуйте интерфейс для формирования заказа и печати штрих-кода.

По нажатию на элемент интерфейса открывается окно формирования заказа, в котором первое, что необходимо – это ввести номер заказа.

Реализуйте поле для ручного ввода номера заказа, при этом номер заказа должен появиться в виде подсказки в поле ввода – последний номер заказа в БД+1. Сотрудник точки проката может подтвердить его нажатием Enter или ввести другой номер, не дублирующий уже существующий в БД, кроме заказов в архиве. По нажатию на Enter (кроме подтверждения номера-подсказки) формируется штрих-код и сохраняется в формате pdf для печати на клейкой бумаге.

Когда сотрудник точки проката формирует штрих-код для заказа, штрих-код должен содержать в себе цифры из уникального идентификатора заказа, даты и времени его создания и срока проката в часах + уникальный код из 6 символов (например: 5 1 4 0 5 2 0 2 2 1 2 1 0 6 1 2 3 4 5 6). Штрих-код должен быть представлен в числовом виде и графически (в виде полосок). Сгенерированный штрих-код должен быть сохранен в pdf файл, так как сотрудник точки проката будет печатать его на клейкой бумаге. Номинальные размеры символа штрихового кодирования:


- высота символа - 25,93 мм;
- высота штриха - 22,85 мм;
- свободная зона слева - 3,63 мм;
- свободная зона справа - 2,31 мм;
- штрихи, которые образуют правый и левый ограничивающие знаки, а также центральный ограничивающий знак должны быть удлинены вниз на 1,65 мм.
- в символе номинального размера высота цифр - 2,75 мм;
- минимальный размер от верхнего края цифр до нижнего края штрихов - 0,165 мм;
- каждая цифра отражается в виде черной полоски, которая имеет ширину $0,15\text{мм} * \{\text{цифра}\}$. Например, если цифра 5, то ширина полоски будет $0,15 * 5$ мм;
- если цифрой является ноль, то необходимо оставлять белый штрих в 1.35мм.
- между штрихами должно быть расстояние в 0.2мм



Цветовое исполнение: возможные цвета штрихов черный, синий, темно-зеленый, темно-коричневый; цвет пробелов (и фона) белый, желтый, оранжевый, светло-коричневый. Не допускаются красный и желтый цвета для штрихов – они не считываются сканером.

После ввода номера заказа и печати штрих-кода в окне формирования заказа реализуйте возможность выбора клиента и услуги.

Если клиента в базе данных нет, его необходимо добавить. Для этого открывается дополнительное модальное окно добавления клиента, в котором нужно ввести все необходимые данные: код клиента, e-mail, ФИО, адрес, дата рождения, серия и номер паспорта, телефон.

Реализуйте возможность добавления нескольких услуг к заказу, поле для добавления новой услуги появляется по нажатию кнопки .

Реализуйте поиск клиента и услуг при создании заказа по всем доступным данным.

Реализуйте возможность просмотра всех клиентов из базы данных.

При формировании заказа стоимость рассчитывается автоматически и формируется электронный вид заказа (дата заказа (системное время), код клиента, номер заказа, ФИО клиента, адрес, перечень услуг, стоимость) с выгрузкой в PDF.

Для удобства передачи, необходимо дополнительно генерировать ссылку с информацией о заказе в следующем формате:

[https://wsrussia.ru/?data=base64\(дата_заказа=2022-04-12T12:12:12&номер_заказа=5\)](https://wsrussia.ru/?data=base64(дата_заказа=2022-04-12T12:12:12&номер_заказа=5))
и сохранение этой ссылки в текстовом документе.

То есть необходимо все данные перевести в формат Base64.

Сессия 2

Модуль 4: Разработка desktop-приложений

Реализация отчетов точки проката горнолыжного курорта «Игора»

Реализуйте интерфейс для построения отчетов.

Реализуйте отчет по оказанным услугам:

- количество оказанных услуг по дням за период времени,
- количество заказов по дням за период времени по каждой услуге,
- количество заказов по дням за период времени.

Пользователь должен выбирать период с точностью до дня.

Все отчеты могут быть представлены в виде графика и таблицы, кроме того, пользователь может выбрать форму представления отчета: графиком или таблицей. Пользователь должен иметь возможность выгрузить данные отчеты в .pdf файл, выбрав один из трех вариантов выгрузки: только график, только таблица, график и таблица.

Разработка библиотеки

Разработайте библиотеку (отдельным проектом), которая позволит вернуть список свободных временных интервалов (заданного размера) в графике сотрудника точки проката горнолыжного курорта «Игора». Это необходимо для понимания того, сколько сотрудников должно быть на точке проката для уменьшения очередей в пиковые периоды, например, выходные или каникулярные дни.

Необходимо обязательно следовать правилам именования библиотек, классов и методов в них. В случае ошибок в рамках именования ваша работа не может быть проверена и ваш результат не будет зачтен. Классы и методы должны содержать модификатор public (если это реализуемо в рамках платформы), чтобы внешние приложения могли получить к ним доступ.

Вход:

- список занятых промежутков времени (в двух массивах: startTimes - начало, durations - длительность),
- минимальное необходимое время для консультации/оформления заказа (consultationTime),
- рабочий день сотрудника (начало - beginWorkingTime и завершение - endWorkingTime)

Выход:

- список подходящих свободных временных промежутков (в массив строк формата HH:mm-HH:mm)

Требования к именованиям и форматам:

	C#	Java	Python
Библиотека классов	SF2022User{NN}Lib.dll	SF2022User{NN}Lib.jar	SF2022User{NN}Lib
Название класса	Calculations	Calculations	Calculations
Название метода	AvailablePeriods()	availablePeriods()	available periods()
Входящие обязательные параметры	TimeSpan[] startTimes, int[] durations, TimeSpan beginWorkingTime, TimeSpan endWorkingTime, int consultationTime	LocalTime[] startTimes, int[] durations, LocalTime beginWorkingTime, LocalTime endWorkingTime, int consultationTime	time[] start times, int[] durations, time begin working time, time end working time, int consultation time
Возвращаемые параметры	string[]	string[]	string[]

Пример:

Вход	Выход
startTime duration	08:00-08:30
10:00 60	08:30-09:00
11:00 30	09:00-09:30
15:00 10	09:30-10:00
15:30 10	11:30-12:00
16:50 40	12:00-12:30
Working Times	12:30-13:00
08:00-18:00	13:00-13:30
Consultation Time	13:30-14:00
30	14:00-14:30
	14:30-15:00
	15:40-16:10
	16:10-16:40
	17:30-18:00

Модуль 7: Тестирование программных решений

Модульные тесты

Реализуйте 10 unit-тестов на основе технологии TDD для библиотеки, функционал которой описан ранее. Учитывайте требования к именованию и входным данным, а так же формату выходных данных.

Важно, чтобы тестовые данные в unit-тестах предусматривали различные ситуации. Например, недостаточное время в промежутках между ранее созданными интервалами, либо в начале рабочего дня, либо в конце рабочего дня; различная длительность и т.д.

Тестовая документация

Для выполнения процедуры тестирования прецедента «Формирование заказа» Вам нужно описать пять сценариев. Добавление заказа может быть выполнено успешно, а может быть отклонено согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма. Для описания тестовых сценариев в ресурсах предоставлен шаблон testing-template.docx.

Модуль 6: Создание инсталляторов

Разработка инсталлятора

Разработайте установочный файл для desktop-приложения под ОС Windows.

Модуль 1: Анализ и проектирование требований, бизнес-процессов

Диаграмма деятельности

Разработайте диаграмму деятельности для прецедента «Формирование заказа». Сохраните файл с диаграммой в форматах .vsdx и .pdf.

Проектирование user story

Разработайте user story на основе технологии BDD для прецедента «Формирование отчета». User story предоставьте на проверку отдельным файлом согласно представленного шаблона в ресурсах.