

## **Дисциплина «Программирование»**

### **Практическое задание №10**

**27 апреля – 17 мая**

#### **«Покупатели и заказы»**

Процесс выполнения этого задания состоит из трёх частей:

- 1) реализация программы, согласно описанным в условии требованиям;
- 2) оценивание работ других студентов;
- 3) период «споров».

#### ***Период реализации программы:***

После выдачи задания Вам необходимо выполнить его и загрузить архив (\*.zip) с решением задачи (полностью заархивировать решение, созданное средой разработки) до крайнего срока. Работа должна выполняться студентом самостоятельно. В работе строго запрещается указывать ФИО, а также любую другую информацию, которая может выдать авторство работы. В случае выявления факта деанонимизации работы, работа может быть аннулирована.

#### ***Период взаимного оценивания:***

После окончания срока, отведенного на реализацию программы, начинается период взаимного оценивания. Вам будет необходимо проверить пять работ других студентов, также выполнявших данное задание, согласно критериям оценивания. Проверка осуществляется анонимно: Вы не знаете, чью работу Вы проверяете, также, как и человек, кому принадлежит решение, не знает, кем была проверена его работа. Помимо оценки Вам необходимо указать комментарий к каждому из критериев. В случае, если Вы снижаете балл, необходимо подробно описать, за что именно была снижена оценка. Также рекомендовано писать субъективные комментарии, связанные с тонкостями программной реализации, предлагать автору работу более оптимальные на ваш взгляд решения. Снимать баллы за субъективные особенности реализации запрещено.

#### ***Период споров:***

По окончании периода взаимного оценивания, в случае если Вы не согласны с оценкой, выставленной одним из проверяющих, Вы можете вступить с этим студентом в анонимный диалог с целью уточнения причин выставления оценки по тому или иному критерию. В случае, если проверяющий не ответил Вам, или вы не пришли к обоюдному решению об изменении оценки, Вы можете поставить флаг, и работа будет рассмотрена одним из преподавателей.

Флаги, поставленные без предварительного обсуждения с проверяющим или после окончания периода выставления флагов, будут отклонены.

Также допустима перепроверка Вашей работы преподавателем или ассистентом. В таком случае оценки других проверявших работу не учитываются.

**Дедлайн загрузки работы:** 17 мая 23:59

**Дедлайн проверки:** 20 мая 23:59

**Дедлайн обсуждения оценок:** 22 мая 23:59

**Дедлайн выставления флагов:** 24 мая 23:59

**Возможность поздней сдачи работы в этом задании предоставляться не будет.**

**Оценивание:**

$O_{\text{итог}} = 0,8 * O_{\text{задание}} + 0,2 * O_{\text{проверки}}$ , где  $O_{\text{задание}}$  – неокруглённая десятибалльная оценка за решение задания выставленная проверяющими с учётом возможной перепроверки преподавателем, а  $O_{\text{проверки}}$  неокруглённая десятибалльная оценка, выставленная студентами, чьи работы вы проверяли. Также в случае, если преподаватель обнаружит, что Вы проверили работы некачественно к Вам могут быть применены санкции в виде штрафа до 3 десятичных баллов от  $O_{\text{итог}}$  (в таком случае оценка вычисляется по формуле  $O_{\text{итог}} = O_{\text{задание}} - \text{Штраф}$ ).

**Задание:**

В прошлом задании был разработан каталог товаров (разделы + товары), а в этом предстоит совершить следующий шаг – реализовать механизм работы с заказами. Однако, не стоит считать, что те, кто не сделал прошлое задание сразу лишаются возможности написать текущее. Минимальный функционал, который должен быть реализован и доступен из прошлого задания - возможность выбора товара из линейного списка товаров (заметьте, ни упоминания про разделы товаров!).

Итак, для реализации механизма работа с заказами необходимо разработать следующие сущности:

- клиент (ФИО, телефон, адрес, e-mail (он же логин), пароль [в открытом виде]);
- заказ (содержит перечень заказанных товаров, включая количество и цену, зафиксированную на момент оформления заказа, а также: номер заказа (уникальное натуральное число), дату+время оформления, статус и, конечно же, ссылку на клиента, оформившего заказ). Статус заказа – перечисление, содержащее следующие значения статусов: обработан,

оплачен, отгружен, исполнен (следует понимать, что каждое из значений статуса – это флаг, который может быть возведен или нет; т.е. может быть заказ в статусе “обработан + оплачен”).

Разработанное приложение должно позволять клиенту:

1. зарегистрироваться или авторизоваться, если он уже зарегистрирован;
2. выбрать товары для заказа (сформировать корзину) и оформить заказ (при оформлении заказу присваивается уникальный номер, а статус не содержит никаких значений);
3. просмотреть список оформленных заказов и их статусов;
4. “оплатить” заказ (учтите, что оплатить можно только заказ имеющий статус “обработан”), добавив значение “оплачен” в статус заказа.

Разработать приложение для продавца (отдельное, или единое – для продавца e-mail можно фиксировать – решать вам), позволяющее:

5. зарегистрироваться или авторизоваться, если он уже зарегистрирован;
6. Просматривать список клиентов (при выборе клиента должна быть возможность просмотра списка всех заказов клиента и отображения оплаченной суммы по всем заказам);
7. Просматривать список всех заказов;
8. Выбрать заказ и изменить его статус (добавить в статус заказа значения: “обработан”, “отгружен”, “исполнен”).
9. Просматривать список активных заказов (активным считается заказ, не имеющий статуса “исполнен”).
- 10.[Дополнительный функционал] Все знают, что хранить пароли клиентов в открытом виде – плохо (хотя такое себе могут иногда позволить и в Facebook). Поэтому, чтобы быть лучше, надо вместо пароля хранить значение хеш-функции (например, SHA-256) от строки (пароль+соль). Соль можно генерировать случайно и хранить в свойствах клиента, а можно использовать в качестве соли существующее поле, например, ФИО – решать Вам.
- 11.[Дополнительный функционал] Реализовать для продавца отчет, позволяющий вывести клиентов, оплативших заказы на сумму, превышающую заданную продавцом + саму эту сумму. Клиенты должны быть упорядочены по убыванию потраченной на заказы суммы.
- 12.[Дополнительный функционал] Иногда производители отзывают свой товар по причине заводского брака, и продавец должен обзвонить всех клиентов, которым отгрузили этот товар. Вам необходимо реализовать отчет, который будет выводить список клиентов, заказавших заданный продавцом товар и дату оформления соответствующего заказа.

Конечно же, все, что делает пользователь/продавец в программе должно автоматически сохраняться в любом удобном вам формате (например, XML, JSON, TXT, BINARY, etc...), чтобы пользователь/продавец мог не

беспокоиться о сохранности данных при завершении работы с программой. А при повторном видеть, что все внесенные им изменения сохранены.

Не забывайте, что если для представления списков пользоваться `IEnumerable<T>`, а для получения результатов запросов удобно применять LINQ.

#### **Дополнительные требования (критерии оценивания):**

1. Для проверки в систему PeerGrade должен быть загружен архив с решением. Ожидается, что проверка работы будет проводиться, в среде разработки Microsoft Visual Studio 2019, поэтому в случае выполнения задания с использованием другой среды разработки настоятельно рекомендуется проверить возможность открытия и запуска проекта в MS VS 2019.
2. Программа не должна завершаться аварийно (или уходить в бесконечный цикл) при любых входных данных. При некорректных входных данных программа должна выводить сообщение об ошибке и запрашивать ввод заново.
3. Программа должна быть декомпозирована. Каждый из логических блоков должен быть выделен в отдельный метод. Не строго, но желательно, чтобы каждый метод по длине не превышал 40 строк.
4. Интерфейс программы должен быть понятен. Пользователю должны выводиться подсказки о возможных дальнейших действиях и иные необходимые сообщения. Предполагается, что для успешного использования программы не требуется обращения к исходному коду программы.
5. Текст программы должен быть документирован. Необходимо писать, как комментарии перед методами, так и комментарии, поясняющие написанный внутри метода код. Названия переменных и методов должны быть на английском языке и отражать суть хранимых значений / выполняемых действий.
6. Работа должна быть выполнена в соответствии с заданием, критерии оценивания (“Rubric”) доступны для предпросмотра в системе «PeerGrade». Критерии оценивания, опубликованные в системе «PeerGrade» являются неотъемлемой частью данного задания.
7. Для получения отличной оценки более 8 баллов студенту необходимо реализовать дополнительный функционал, описанный в п.10 – п.12 задания.
8. Также оценивается общее впечатление, которое производит работа (как с точки зрения пользовательского интерфейса, так и с точки зрения написания кода программы). Эта часть оценки выставляется по усмотрению проверяющего.