

Дисциплина «Программирование»

Практическое задание №9

23 марта – 12 апреля

«Склад товаров»

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: 12 апреля 23:59

Дедлайн проверки: 15 апреля 23:59

Дедлайн обсуждения оценок: 17 апреля 23:59

Дедлайн выставления флагов: 18 апреля 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:

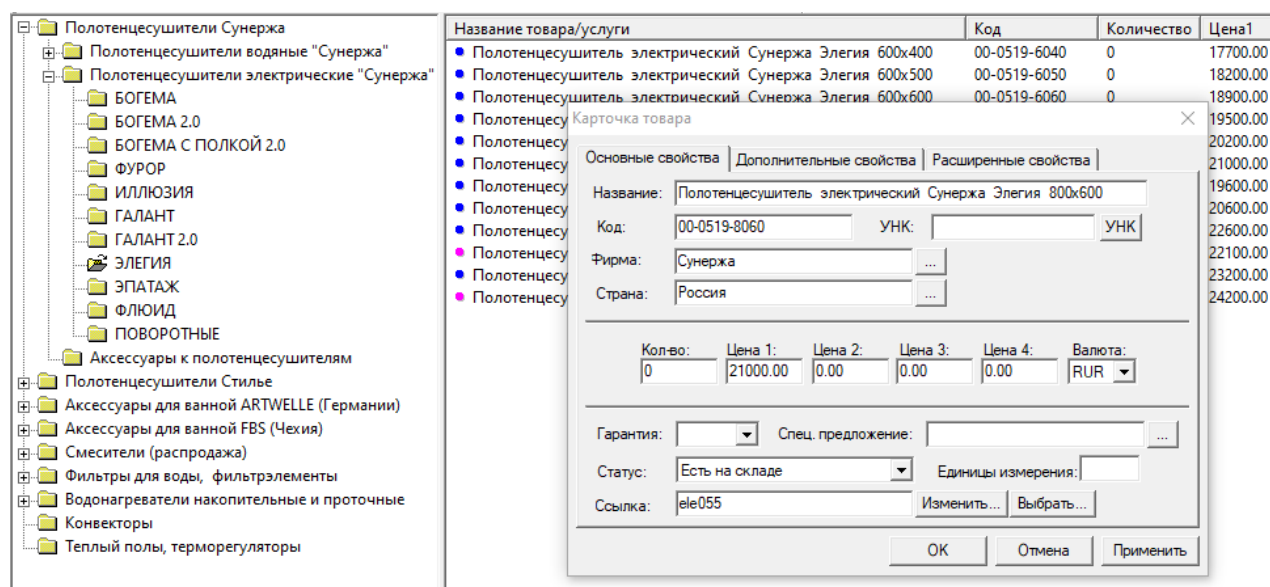
Разработать приложение с графическим пользовательским интерфейсом Windows Forms(должно работать на одной из перечисленных платформ: .NET 5, .Net Core 3.1 или .Net Framework 4.7.2.), позволяющее управлять обширной номенклатурой товаров, хранимых на складе.

Поскольку на складе могут храниться самые разные товары, было принято решение об обязательной их классификации для удобства. Самый удобный классификатор, это, конечно же, древовидная структура, где каждый узел – раздел, который описывается, как минимум, названием и может содержать товары. Учтите, что в классификаторе не может быть двух одноименных разделов, имеющих общего родителя. На глубину дерева классификатора не накладывается никаких ограничений.

Каждый товар склада обязан принадлежать только одному разделу (не может быть товаров, не привязанных к разделу классификатора). Сам товар, конечно же, описывается целым набором обязательных (наименование, артикул, цена, целочисленное количество – остаток на складе) и дополнительных

характеристик (текстовое описание, фотография).

Конечно же, все, что делает пользователь в программе должно автоматически сохраняться в любом удобном вам формате (например, XML, JSON, TXT, BINARY, etc...), чтобы пользователь мог не беспокоиться о сохранности данных при завершении работы с программой. А при повторном входе, что все внесенные им изменения сохранены. Пример фрагмента интерфейса программы показан на рисунке.



Программа должна обладать следующей функциональностью:

1. Вывод и управление классификатором, включая: создание разделов классификатора, изменение, удаление. Позаботиться о запрете удаления непустого раздела (т.е. раздела, содержащего другие подразделы или товары).
2. Вывод списка товаров в разделе и управление товарами с обязательным набором характеристик, включая: создание товара (конечно, с привязкой к разделу классификатора), изменение, удаление.
3. Сохранение состояния склада (классификатор + товары) в файл (или набор файлов) и загрузка состояния склада из файла (или набора файлов) по запросу пользователя. Таким образом, появится возможность управления несколькими складами из программы.
4. Предусмотреть возможность автоматического создания CSV-отчета (пригодного для загрузки в Excel), в котором содержится список товаров, заканчивающихся на складе и поэтому требующих дозаказа у поставщиков (т.е. товаров, количество которых меньше заданного пользователем приложения значения), содержащего следующие столбцы: <путь_классификатора>, артикул, название товара, количество на складе. Здесь <путь_классификатора> - разделенный слешами список разделов классификатора от корня до раздела, в

котором размещен товар, например, “Полотенцесушители Сунержа / Полотенцесушители электрические "Сунержа" / ЭЛЕГИЯ”.

5. Скрывать или делать неактивными компоненты управления (интерфейса пользователя), когда их невозможно использовать.
6. [Дополнительный функционал] Реализовать возможность отображения не только списка товаров, расположенных в выбранном разделе классификатора, но и товаров, расположенных во всех подразделах данного раздела (т.е. “ниже” в классификаторе). Товары из подразделов необходимо выделять в списке, чтобы иметь возможность отличить их от товаров расположенных непосредственно в текущем разделе.
7. [Дополнительный функционал] Предусмотреть возможность автоматической (используем Random) генерации классификатора и товаров по заданным пользователям параметрам, включающим, как минимум: количество разделов классификатора и количество товаров.
8. [Дополнительный функционал] Добавить к разделу классификатора свойство “код сортировки” (и соответственно функционал по управлению им), позволяющий упорядочить разделы классификатора в соответствии с установленным кодом сортировки, а при равенстве кодов – по названию раздела лексикографически.
9. [Дополнительный функционал] Поддержка дополнительных характеристик товара (текстовое описание, фотография) с возможностью просмотра изображения. Учтите, что при сохранении состояния склада (п.3) требуется сохранять и изображение для товара (например, в непосредственно файле в кодировке BASE64 или отдельно копировать файл изображения в папку).

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 пред-просмотра в системе “PeerGrade”.
7. Для получения отличной оценки более 8 баллов студенту необходимо реализовать дополнительный функционал, описанный в п.6 – п.9 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.