**Лабораторная работа №6**

**Марченко и Фесько**

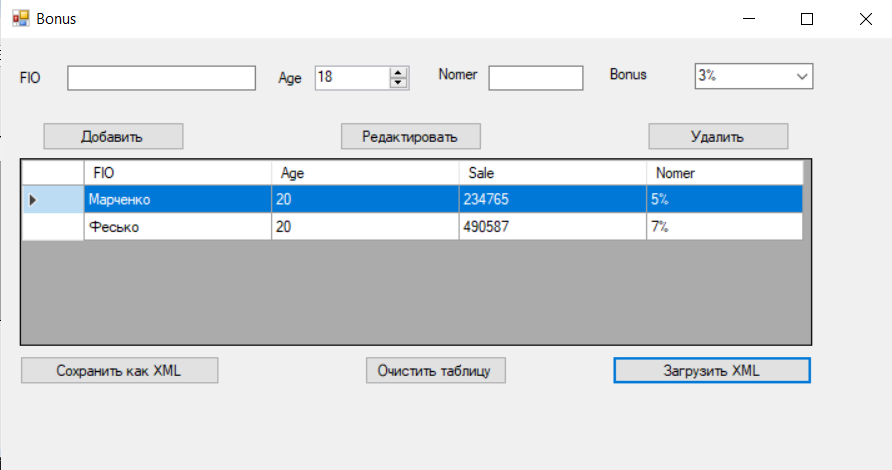
**Группа 1720**

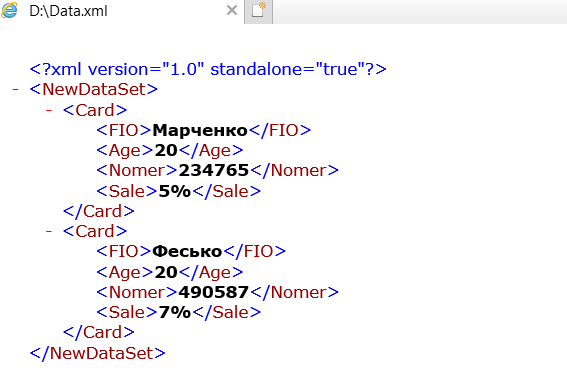
Программа представляет из себя приложение, работающее с файлами и xml-данными.

Представлена система ведения бонусной программы. Реализована регистрация новых бонусных карточек, с хранением данных в xml-формате. А также возможность изменения процента скидки.

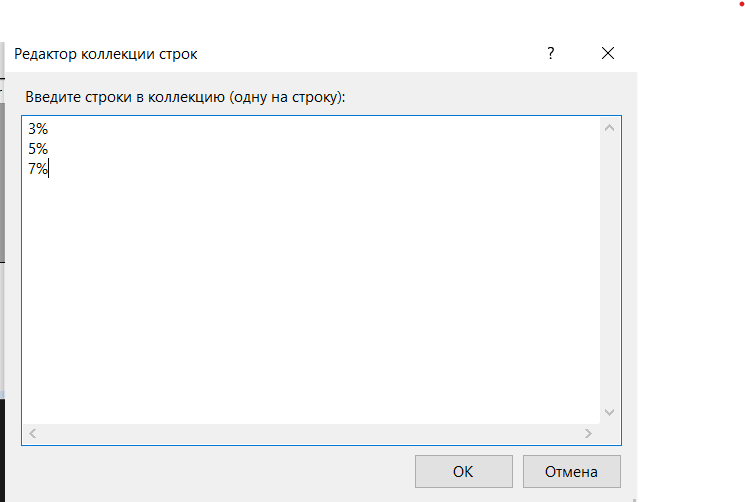
В программе реализовано несколько методов: добавление данных в xml-документ, редактирование, удаление, а также возможность загрузки документа в таблицу приложения.

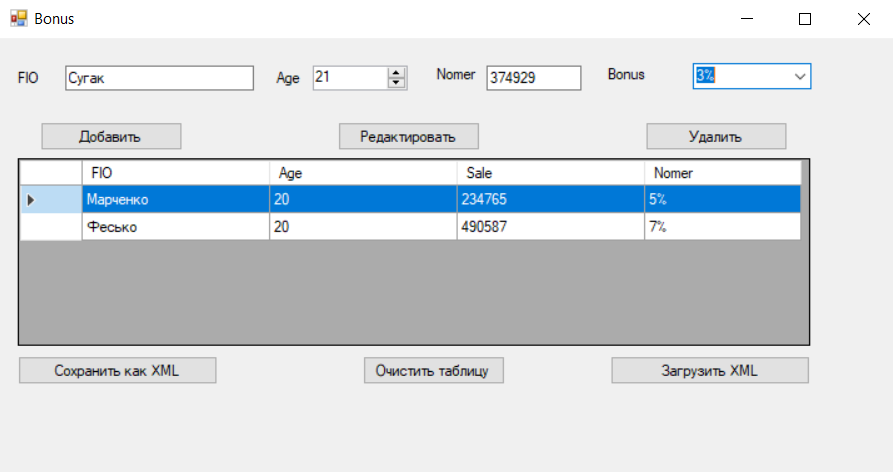
На скриншоте представлена форма приложения. С помощью кнопки «Загрузить XML», мы загружаем данные из документа в таблицу приложения для наглядности. Примечание: при загрузки файла, в приложение, таблица (dataGridView1), должна быть очищена с помощью кнопки «Очистить таблицу».



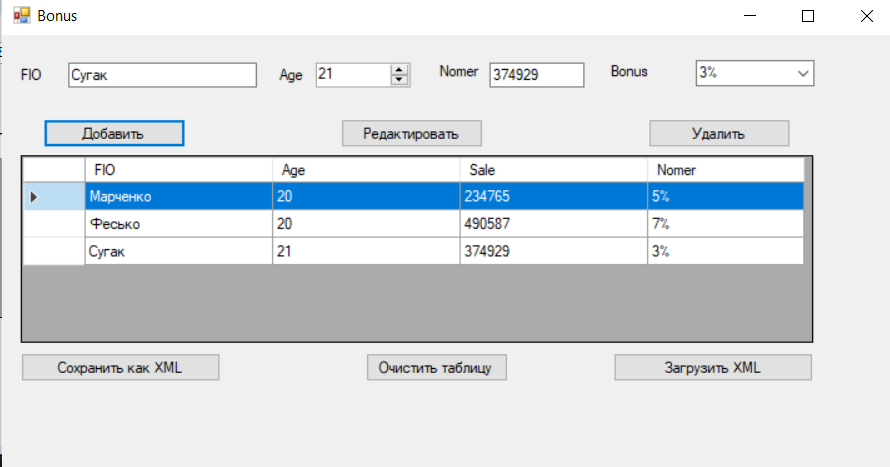


В форме доступны поля, в которые необходимо вносить информацию. Возраст вводится с помощью элемента numericUpDown1. Номер бонусной карты должен содержать 6 цифр. И бонусный процент выполнен с помощью коллекции, включающая в себя: 3%, 5%, 6%.

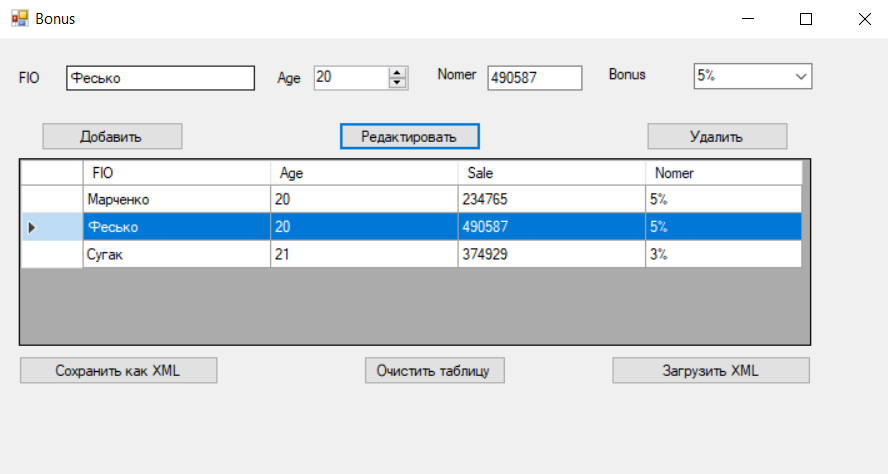




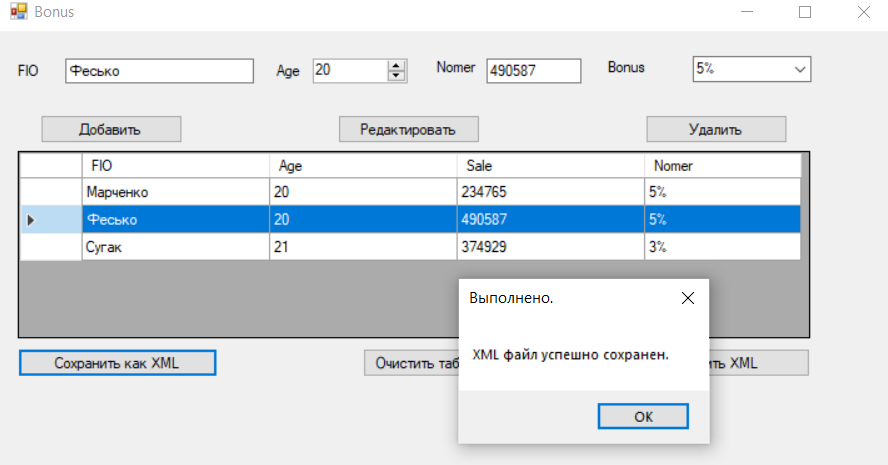
Введя данные нового покупателя «Сугак» и, при нажатии кнопки «Добавить», данные добавляются в таблицу.

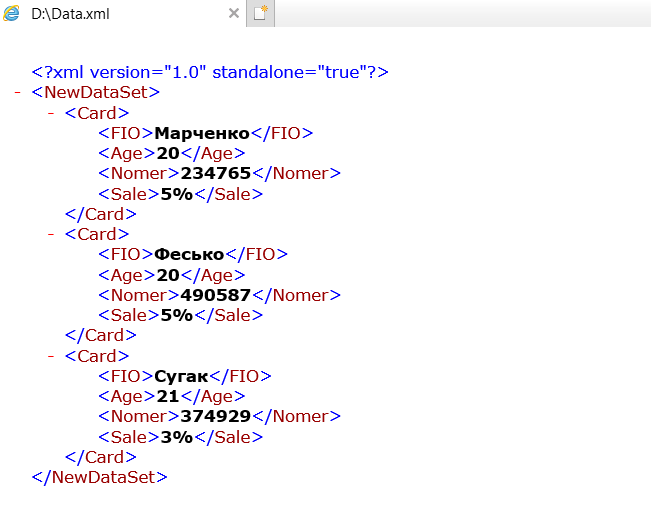


Чтобы редактировать запись, необходимо нажать на строку с нужным пользователем, в верхних полях отображаются данные, которые можно изменить, например поменяем скидку с 7% на 5%. И нажав кнопку «Редактировать», данные в таблице поменяются.

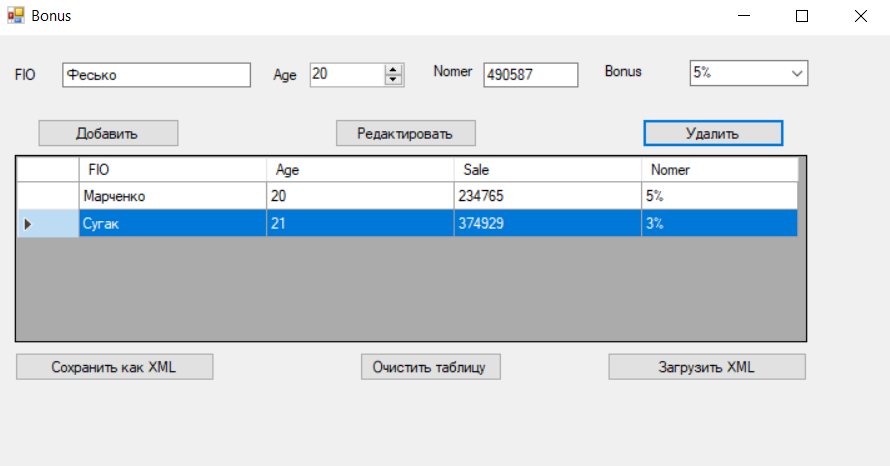


При нажатии кнопки «Сохранить как XML», данные заносятся в файл.





При выборе пользователя в таблице также можно удалить его из таблицы, нажав кнопку «Удалить». Затем, чтобы изменить данные в файле, необходимо его ещё раз сохранить.



Данная лабораторная работы выполнялась в паре.

Изначально без репозитория.

Но мы разобрались с работой в команде с помощью GitHub.

Для упрощения работы, в Visual Studio предусмотрена возможность работы с GitHub.

Для этого, сначала необходимо было выполнить некоторые действия.

Если проект не загружен в гит, в правом нижнем углу должна быть кнопка «Добавить в систему управления версиями».



Прописав свои данные и создав репозиторий, мы смогли пользоваться данными функциями.

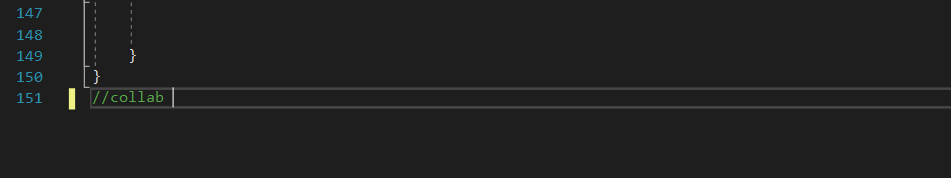
В гит можно создать команду и добавить соавторов с помощью коллаборации, чтобы можно было делать проект совместно и делиться изменениями.

Если пользоваться командной строкой, то изменения выполняются с помощью ряда команд: чтобы забросить файлы с изменениями используются git add, git commit -m "message", git push, чтобы получить файлы используется команда git pull. Чтобы посмотреть все команды, можно написать git, и покажется список всех команд с описанием. Также есть функция pull request, с помощью которой, можно показать свои изменения, и дать участникам команды проверить код, чтобы в дальнейшем сохранить хороший исходник. Это некий специальный форум для обсуждения.

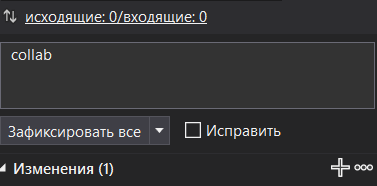
Добавление изменений в visual studio выполняются очень просто.

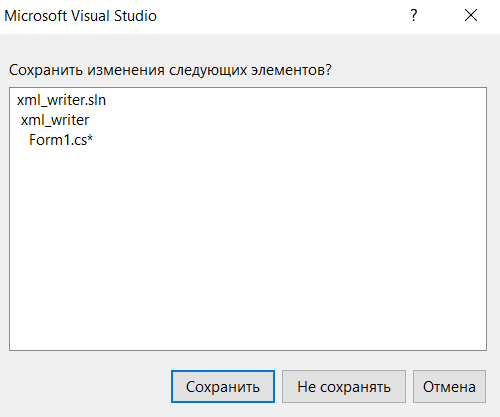
Далее на примере будет описана команда commit.

Напишем комментарий в конце файла.



Далее необходимо написать описание к изменению. Нажать на кнопку «Зафиксировать всё», будет предложено сохранить изменения, нажимаем сохранить.

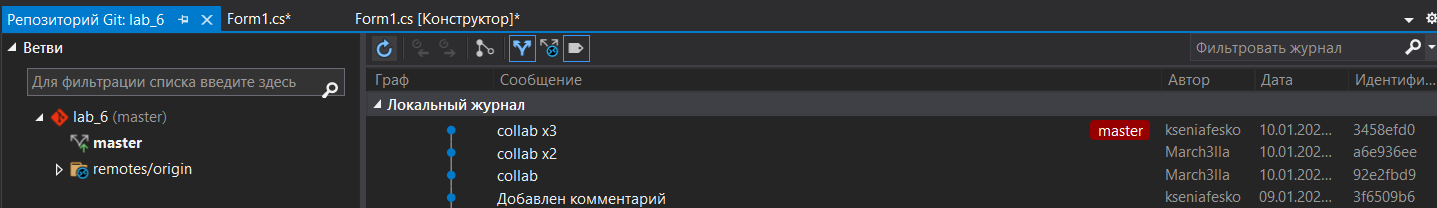


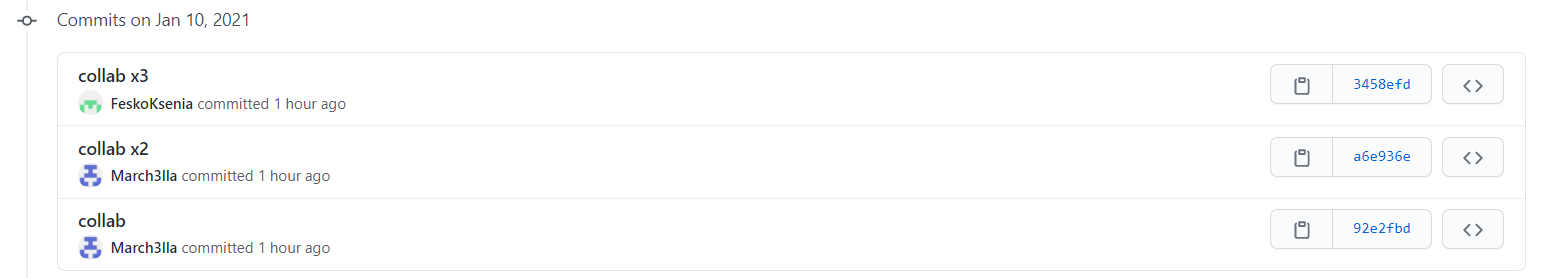


Затем, отправляем изменения в репозиторий, с помощью кнопки отравить в правом верхнем углу, изменения отображаются в локальном журнале или на сайте гитхаба.







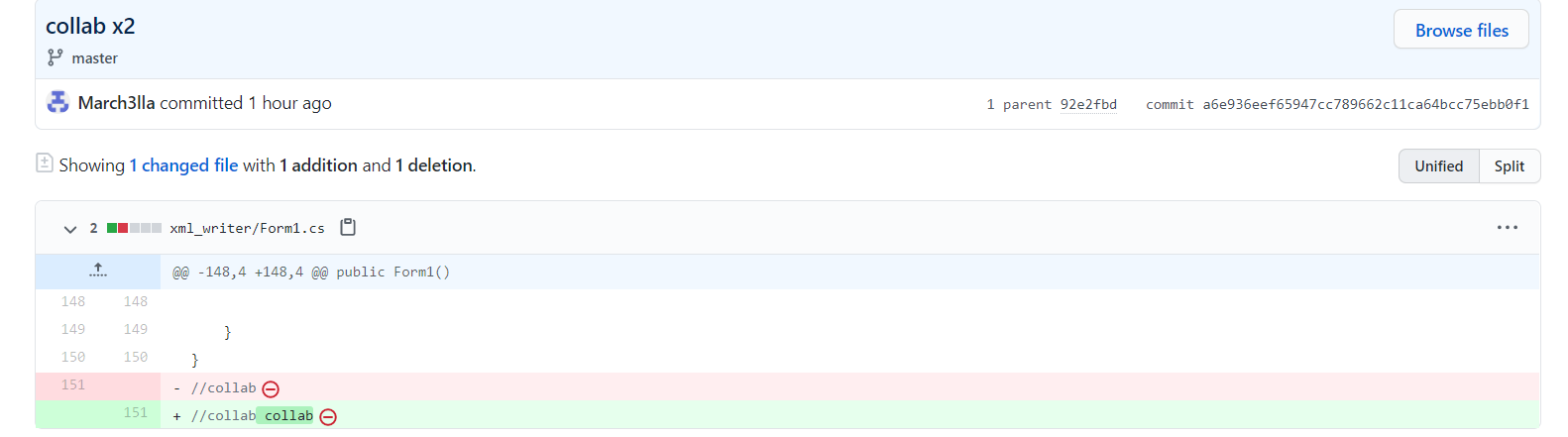


С помощью кнопки «Вытянуть»



Наш файл обновляются и показываются последние внесённые изменения.

На сайте гитхаб изменения отображаются таким образом, нажав на интересующее нас изменение, показывается строка, которая была и новая строка.



**Вопросы для самопроверки**

1. **Что представляет из себя .NET?**

Платформа .NET Framework — это технология, которая поддерживает создание и выполнение веб-служб и приложений Windows. Платформа .NET Framework состоит из общеязыковой среды выполнения (среды CLR) и библиотеки классов .NET Framework. Основой платформы .NET Framework является среда CLR. Платформа .NET образует каркас, включающий библиотеку классов, технологии для доступа к данным и построения оконных и веб-приложений.

1. **Что такое CLR и какие основные задачи она выполняет?**

Основное назначение CLR (Common Language Runtime) — выполнение приложений, соблюдение всех программных зависимостей, управление памятью, обеспечение безопасности, интеграция с языками программирования.

Основная задача CLR – это манипулирование сборками: загрузка, JITкомпиляция, создание окружения для выполнения сборок. Важной функцией CLR является управление памятью при работе приложения и выполнение автоматической сборки мусора, то есть фонового освобождения неиспользуемой памяти. Кроме этого, CLR реализует в приложениях для .NET проверку типов, управление политиками безопасности при доступе к коду и другие функции.

1. **Что такое CTS и каково её предназначение?**

CTS ( Common Type System — общая система типов), которая полностью описывает все типы данных, поддерживаемые средой выполнения, определяет, как одни типы данных могут взаимодействовать с другими и как они будут представлены в формате метаданных .NET. Например, в .NET любая сущность является объектом какого-нибудь класса, производного от корневого класса System.Object. Спецификация CTS поддерживает такие общие понятия, как классы, делегаты (с поддержкой обратных вызовов), ссылочные и размерные типы.

1. **Назовите преимущества .NET.**

Объектно-ориентированное программирование. И среда .NET Framework, и С# изначально полностью базировались на объектноориентированных принципах.

Хороший дизайн. Библиотека базовых классов, которая спроектирована "с нуля", исключительно интуитивно понятным образом.

Независимость от языка. Благодаря .NET, код всех языков, то есть VisualBasic .NET, С#, J# управляемого С++, компилируется в общий язык промежуточного уровня — Intermediate Language. Это значит, что все эти языки обладают возможностями взаимодействия, как никогда ранее.

Лучшая поддержка динамических Web-страниц. Хотя ASP предлагал высокую степень гибкости, он также был и неэффективен из-за своих интерпретируемых сценарных языков, а недостаток объектно-ориентированного дизайна часто приводил к запутанному коду ASP. .NET предлагает интегрированную поддержку Web-страниц с применением новой технологии — ASP.NET. В ASP.NET код ваших страниц компилируется и может быть написан на языке высокого уровня, поддерживающего .NET— таком как С# или Visual Basic 2012.

Эффективный доступ к данным. Набор компонентов .NET, известный под общим названием ADO.NET, предоставляет эффективный доступ к реляционным базам данных и широкому разнообразию других источников данных. Также имеются компоненты, предоставляющие доступ к файловой системе и каталогам. В частности, в .NET встроена поддержка XML, что позволяет манипулировать данными, которые могут быть импортированы их других, не-Windows платформ, и экспортированы на них.

Разделение кода. Среда .NET полностью изменила способ разделения кода между приложениями, введя концепцию сборки (assembly), которая заменила традиционные библиотеки DLL. Сборки имеют форматные средства для указания версий, и одновременно в системе могут существовать разные версии одних и тех же сборок.

Повышенная безопасность. Каждая сборка также может содержать встроенную информацию безопасности, которая в точности описывает, кому и каким категориям пользователей или процессов какие методы каких классов разрешено вызывать. Это обеспечивает очень высокую степень контроля за тем, как могут использоваться сборки, которые вы поставляете.

И другое.

1. **Какие категории типов существуют в C# и в чем их отличие?**

В C# имеются две категории типов: типы значений и ссылочные типы. Разница между ними в том, что тип значения хранит данные непосредственно, в то время как ссылочный тип хранит ссылку на значение. Эти типы сохраняются в разных местах памяти: типы значений хранятся в области, известной как стек, а ссылочные типы – в области, которая называется управляемой кучей.

1. **Что такое пространства имен и какие свойства они имеют?**

Все определяемые классы и структуры, как правило, не существуют сами по себе, а заключаются в специальные контейнеры - пространства имен.

Пространства имен имеют следующие свойства.

− Организация крупных проектов по созданию кода.

− Для их разделения используются оператор.

− Директива using исключает требование на указание имени пространства имен для каждого класса.

− Пространство имен global является корневым пространством имен: global::System всегда будет ссылаться на пространство имен платформы .NET Framework System.

1. **Какие элементы может содержать класс в C#?**

Класс может содержать следующие элементы:

− Поле − это переменная любого типа, которая объявлена непосредственно в классе.

− Константа − это переменная, значение которой нельзя изменить. Константы используются для гарантирования того, что данные в этой переменной не изменятся. Для того, чтобы объявить константу, перед обычным объявлением переменной нужно добавить ключевое слово const.

− Метод представляет собой блок кода, содержащий набор инструкций.

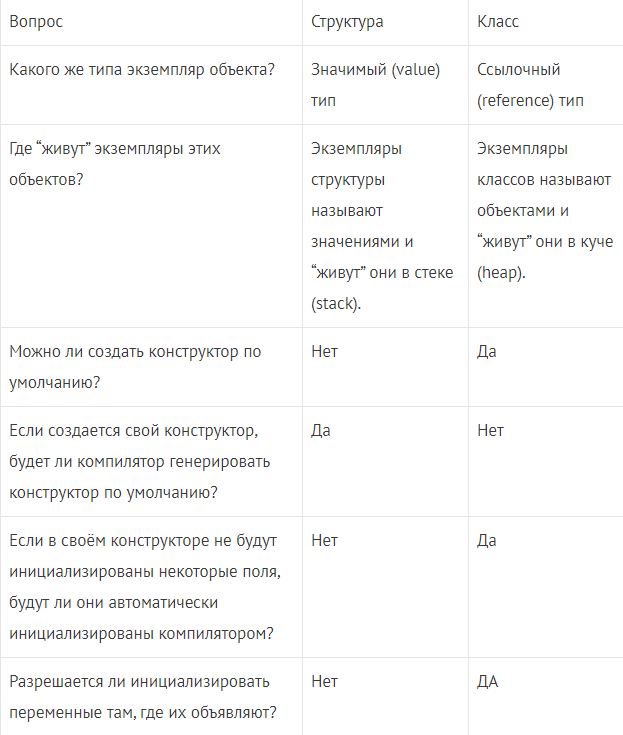
− Свойство − это член, предоставляющий гибкий механизм для чтения, записи или вычисления значения частного (private) поля.

1. **Что такое делегат и как он объявляется?**

Делегат – это пользовательский тип, который инкапсулирует метод. В C# делегат объявляется с использованием ключевого слова delegate.

1. **Чем структура отличается от класса?**

Структуры, указываемые в списке параметров метода, передаются по значению (то есть копируются), объекты классов — по ссылке.



1. **Что такое универсальный шаблон?**

Универсальные шаблоны (generics) позволяют при разработке пользовательского типа или метода указать в качестве параметра тип, который конкретизируется при использовании. Универсальные шаблоны применимы к классам, структурам, интерфейсам, делегатам и методам.

Для описания универсального типа используется следующий синтаксис: после имени класса в угловых скобках указывается параметр типа, допускается использовать несколько параметров-типов.

**11. Какой набор типов включает в себя .NET для предоставления**

**стандартных коллекций?**

Платформа .NET включает большой набор типов для предоставления стандартных коллекций - списков, множеств, словарей. Эти типы можно разделить на несколько категорий: базовые интерфейсы и вспомогательные классы, классы для коллекций-списков и словарей, набор классов для построения собственных коллекций.

1. **Что такое LINQ? Какие существуют типы LINQ?**

LINQ (Language-Integrated Query) представляет простой и удобный язык запросов к источнику данных.

По типу обрабатываемой информации LINQ делится на

* LINQ to Objects – библиотеки для обработки коллекций объектов в памяти;
* LINQ to SQL – библиотеки для работы с базами данных;
* LINQ to XML предназначена для обработки XML-информации.