

Эта часть работы выложена в ознакомительных целях. Если вы хотите получить работу полностью, то приобретите ее воспользовавшись формой заказа на странице с готовой работой:

<https://studservis.ru/gotovye-raboty/kontrolnaya-rabota/179437>

Тип работы: Контрольная работа

Предмет: Программирование (другое)

ВВЕДЕНИЕ 3

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5

ПРАКТИЧЕСКАЯ ЧАСТЬ 10

ЗАКЛЮЧЕНИЕ 15

СПИСОК ЛИТЕРАТУРЫ 16

ПРИЛОЖЕНИЯ 17

ВВЕДЕНИЕ

Объектно-ориентированное программирование – это подход, при котором вся программа рассматривается как набор взаимодействующих друг с другом объектов. При этом нам важно знать их характеристики. У каждого объекта в системе есть свойства и поведение, как и у любого реального объекта. Например, рассмотрим объект «машина». У него есть свойства (цвет, вес, стоимость) и поведение (машина может ехать, сигнализировать, потреблять топливо).

Такой подход помогает строить сложные системы более просто и естественно благодаря тому, что вся предметная область разбивается на объекты и каждый из них слабо связан с другими объектами. Слабая связанность возникает вследствие соблюдения трех принципов: инкапсуляции, наследования и полиморфизма.

Инкапсуляция – сокрытие поведения объекта внутри него. Объекту «водитель» не нужно знать, что происходит в объекте «машина», чтобы она ехала. Это ключевой принцип ООП.

Наследование. Есть объекты «человек» и «водитель». У них есть явно что-то общее. Наследование позволяет выделить это общее в один объект (в данном случае более общим будет человек), а водителя — определить как человека, но с дополнительными свойствами и/или поведением. Например, у водителя есть водительские права, а у человека их может не быть.

Полиморфизм – это переопределение поведения. Можно снова рассмотреть «человека» и «водителя», но теперь добавить «пешехода». Человек умеет как-то передвигаться, но как именно, зависит от того, водитель он или пешеход. То есть у пешехода и водителя схожее поведение, но реализованное по-разному: один перемещается ногами, другой – на машине.

ООП позволяет упростить сложные объекты, составляя их из более маленьких и простых, поэтому над программой могут работать сотни разработчиков, каждый из которых занят своим блоком. Большинство современных языков программирования — объектно-ориентированные, и, однажды поняв суть, вы сможете освоить сразу несколько языков.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В качестве задания для контрольной работы выбран вариант № 2.

Пусть требуется создать программную систему, предназначенную для организаторов выставки собак. Она должна обеспечивать хранение сведений о собаках - участниках выставки и экспертах. Для каждой собаки в БД должны храниться сведения, о том, к какому клубу она относится, кличка, порода и возраст, сведения о родословной (номер документа, клички родителей), дата последней прививки, фамилия, имя, отчество и паспортные данные хозяина. На каждый клуб отводится участок номеров, под которыми будут выступать участники выставки. Сведения об эксперте должны включать фамилию и имя, номер ринга, который он обслуживает; клуб, название клуба, в котором он состоит. Каждый ринг могут обслуживать несколько экспертов. Каждая порода собак выступает на своем ринге, но на одном и том же ринге в разное время могут выступать разные породы. Итогом выставки является определение медалистов по каждой породе. Организатор выставки должен иметь возможность добавить в базу нового участника или нового эксперта, снять эксперта с судейства, заменив его другим, отстранить собаку от участия в выставке. Организатору

выставки могут потребоваться следующие сведения;

- На каком ринге выступает заданный хозяин со своей собакой?
- Какими породами представлен заданный клуб?
- Какие медали и сколько заслужены клубом?
- Какие эксперты обслуживают породу?
- Количество участников по каждой породе?.

Необходимо предусмотреть возможность выдачи стандартной справки о призовом месте, занятом собакой на выставке и отчета о результатах выступления каждого клуба (сколько всего участников, какие породы, медалисты по породам).

Для выполнения задания использовалась язык программирования C# и среда Sharpdevelop 4.

C# - современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать множество типов безопасных и надежных приложений, работающих в экосистеме .NET. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript. Здесь представлен обзор основных компонентов языка C# 8 и более ранних версий. Если вы хотите изучить язык с помощью интерактивных примеров, рекомендуем поработать с вводными руководствами по C#.

C# - это объектно - и компонентно-ориентированный язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО.

Вот лишь несколько функций языка C#, которые позволяют создавать надежные и устойчивые приложения.

*Сборка мусора _ автоматически освобождает память, занятую недоступными неиспользуемыми объектами.

Типы, допускающие значение null, обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них. Лямбда-выражения поддерживают приемы функционального программирования. Синтаксис LINQ создает общий шаблон для работы с данными из любого источника. Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем. В C# действует _ *единая система типов*. Все типы C#, включая типы-примитивы, такие как int и double, наследуют от одного корневого типа object. Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, C# поддерживает как определяемые пользователями ссылочные типы, так и типы значений. C# позволяет динамически выделять объекты и хранить упрощенные структуры в стеке. C# поддерживает универсальные методы и типы, обеспечивающие повышенную безопасность типов и производительность. C# предоставляет итераторы, которые позволяют разработчикам классов коллекций определять пользовательские варианты поведения для клиентского кода.

В C# особое внимание уделяется управлению версиями для обеспечения совместимости программ и библиотек при их изменении. Вопросы управления версиями существенно повлияли на такие аспекты разработки C#, как отдельные модификаторы virtual и override, правила разрешения перегрузки методов и поддержка явного объявления членов интерфейса.

SharpDevelop — свободная среда разработки для C#, Visual Basic .NET, Boo, IronPython, IronRuby, F#, C++.

Обычно используется как альтернатива Visual Studio .NET. Существует также форк на Mono/GTK+ — MonoDevelop.

SharpDevelop 2.0 предоставляет интегрированный отладчик, который использует собственные библиотеки и взаимодействует с исполняющей средой .NET через COM Interop.

Хотя SharpDevelop 2.0 (как и VS2005) использует файлы проекта в формате MSBuild, он по-прежнему может использовать компиляторы от .NET Framework 1.0 и 1.1, а также от Mono.

На текущий момент поддерживается версия .Net Frameworks от 2.0 до 4.5.1.

Схема всех спроектированных классов

Программа состоит из одного класса Dog и 4 структур :

1. struct DogClub – содержит информацию о клубе, кличке, породе и занятых местах участвующих в конкурсе собак.
2. Dog_info – содержит информацию о клубе, кличке, породе, возрасте, информации о родословной, дате последней прививки, ФИО хозяина, паспорте хозяина.

3. Dog_expert – содержит информацию о судьях – клуб, ФИО, номер ринга, породу которую судит эксперт.
4. Dog_performance – содержит информацию о выступлении №, номер ринга, породу, экспертов оценивающих выступление, время, собак которые участвуют в выступлении их номера, клубы, оценки за выступление.

Класс Dog содержит 6 методов :

1. Autocomplete() – при запуске программы автоматически заполняет структуры Dog_info, Dog_expert данными содержащимися в строковых массивах

1. dog_club
2. dog_nickname
3. dog_breed
4. dog_ege
5. dog_pedigree_information
6. dog_date_of_vaccination
7. full_name_master
8. pasport_master
9. Dog_expert_full_name

2. Hold_dog_show() – метод который проводит соревнования. Выводит информацию и генерирует оценки. Также после

1. <https://metanit.com/>
2. <https://habr.com/ru/post/114546/>
3. <https://tproger.ru/experts/oop-in-simple-words/>
4. <https://www.cyberforum.ru/ado-net/thread2200741.html>
5. <https://ru.stackoverflow.com/questions>
6. <https://vscode.ru/prog-lessons/sortirovka-list-v-c-sharp.html><https://vscode.ru/prog-lessons/sortirovka-list-v-c-sharp.html>

Эта часть работы выложена в ознакомительных целях. Если вы хотите получить работу полностью, то приобретите ее воспользовавшись формой заказа на странице с готовой работой:

<https://studservis.ru/gotovye-raboty/kontrolnaya-rabota/179437>