

Задание

Вариант 1

Реализуйте программу с интерфейсом на WinForms/WPF, которая позволяет:

1. Считать данные о футбольных игроках из файла FIFA.csv.

Каждая строка представляет информацию об одном игроке. В таблице с данными хранятся следующие параметры игроков:

1. sofifa_id - unique number on sofifa
2. player_url - url of the scraped player
3. short_name - short name of the player
4. long_name - long name of the player
5. age - age of the player
6. dob - date in which the player was born
7. height_cm - height in cm of the player
8. weight_kg - weight in kg of the player
9. nationality - nationality of the player
10. club - club of the player
11. overall - overall attribute of the player
12. potential - potential attribute of the player

NB: для работы с файлом FIFA.csv необходимо реализовать собственный парсер. Встроенную реализацию CsvReader и пр. использовать нельзя.

2. Вывести информацию из файла на экран приложения с возможностью ее редактирования. Рекомендуется использовать [DataGridView](#).
3. Отфильтровать информацию на экране приложения по следующим параметрам игроков:
overall, potential, nationality.
4. Запустить процесс сражения между двумя игроками, где Игрок_2 - компьютер:
 1. Игрок_1 должен иметь возможность по выведенной информации из файла набрать команду из 11 футбольных игроков.
 2. Игрок_2 должен иметь возможность по выведенной информации из файла набрать команду из 11 футбольных игроков, не выбранных Игроком_1.
 3. Процесс сражения делится на раунды. За один раунд оба игрока выполняют следующие действия:
 - i. Игроки производят процесс нападения:
 1. Игрок_1 выбирает участника своей команды, которым он производит атаку;
 2. Игрок_2 выбирает участника своей команды, которым он производит защиту;
 3. Расчёт результатов по формуле: $((height_cm - weight_kg) / 10) * overall / \max(overall - potential, 1)$ – производится расчёт очков для обоих выбранных игроков. Кто набрал больше очков, тот выиграл этап:
 - a. Если выиграл нападающий игрок, то идет повтор этапа;
 - b. Если выиграл игрок защиты, то раунд заканчивается.
 - ii. Если игрок прошел 4 этапа подряд, он забивает гол и раунд заканчивается.
 - iii. Игра длится 30 раундов. Каждый раунд атакующий и защитник меняются местами.
 - iv. По окончании игры побеждает игрок с наибольшим количеством забитых голов.
 5. Реализовать возможность автоматического сохранения текущего состояния игры после каждого раунда в JSON файл, разработанного Вами формата. При запуске программы необходимо предлагать пользователю продолжить последнюю автоматически сохраненную игру (если такая есть) или начать новую игру

Замечания:

1. Необходимо соблюдать все изученные принципы ООП.
2. Необходимо соблюдать декомпозицию (включая разделение классов по файлам и создания библиотек классов).
3. Явной архитектуры и спецификации нет, необходимо продумать самому то, как будет устроена ваша программа. Рекомендуется сперва попробовать изобразить работу программы на рисунке.

Для особо желающих:

1. Изучив LINQ, часть задач из ДЗ могут решиться проще.

Замечания:

4. Необходимо соблюдать инкапсуляцию
5. В классах можно добавлять свои свойства для доступа к полям, методы и поля
6. Можно добавлять свои классы/члены классов
7. Спецификацию полей и методов типов из условия менять нельзя

Ограничения и требования:

1. Предусмотреть цикл повторения решения
2. Использовать конструкцию **try catch** в местах, где могут возникнуть исключения. Максимально конкретизировать реакцию программы по типам исключений.

Существенные требования:

1. Программа должна компилироваться.
2. Входные данные должны обрабатываться и не порождать исключительные ситуации.
3. Не изменять спецификацию указанных в задании нестатических методов.

Требования к интерфейсу:

Интерфейс должен быть реализован на WindowsForms или WPF (только .Net Framework).