Лабораторная работа №1 МДК 01.01.

Гаврильченко Степан П-30

Вариант №2

№1 Описание программных функций

1. Создание списка с информацией о футболистах
2. Ввод запроса пользователя по названию команды
3. Поиск всех футболистов по названию команды
4. Сортировка футболистов команды по количеству забитых мячей
5. Поиск футболистов с наивысшим количеством мячей. Результат – список топ 3 футболиста команды с наивысшим количеством забитых мячей
6. Вычисление разницы баллов по сравнению с количеством баллов команды, занявшей первое место
7. Вывод на экран списка топ 3 игрока по количеству забитых мячей и разницу баллов по сравнению с командой, занявшей первое место

№2 Описание структуры информации

1. Сведения о футболисте:

String: Имя

Int: число забитых мячей

Class Player

1. О команде футболистов:

List<Player> Players;

String: название команды

Int: количество набранных очков

Class Team

1. Конструкторы для классов Player и Team, обозначение переменных

Public Player (string name, int goals)

{

Name = name;

Goals = goals;

}

Public Team (string name, int points, List<Player> players)

{

Name = name;

Points = points;

Players = players;

}

1. Функция показа топа бомбардиров

Main() — главная функция, которая запускает программу и выполняет:

1. Загружает список команд (GetTeams()).
2. Запрашивает у пользователя название команды.
3. Ищет команду (FindTeam()).
4. Если команда найдена:
   * Выводит топ-3 бомбардиров (GetTopScorers()).
   * Считает, насколько команда отстает от лидера (GetMaxPoints()).
5. Если команда не найдена — сообщает об ошибке.

List<Team> GetTeams()**— метод, который загружает спискок команд.**

* Возвращает **названия команд** с игроками

Team FindTeam(List<Team> teams, string teamName) — метод, который осуществляет поиск команды.

* teams — список всех команд.
* teamName — введенное пользователем название.

Работа по следующему принципу:

* Проверяет каждую команду в списке.
* Если название совпадает (без учета регистра) — возвращает команду.
* Если не находит — возвращает null(ничего).

List<Player> GetTopScorers(List<Player> players) — метод, котрый находит топ-3 бомбардира.

* players — список игроков команды.
* Возвращает 3 игрока с наибольшим числом голов.

Работа по следующему принципу:

1. Сначала добавляет первых игроков, пока не наберется 3.
2. Затем для каждого нового игрока:
   * Если у него больше голов, чем у кого-то из топ-3 — заменяет самого слабого.
3. В конце сортирует топ-3 по убыванию голов.

Void PrintTopScorers(List<Player> topScorers)**— метод, который выводит на экран топ-3 бомбардира**

Int GetMaxPoints(List<Team> teams) — метод, который находит максимальное максимальное число очков

* teams — список всех команд.
* Возвращает наибольшее количество очков среди всех команд.

Int CalcPointsDifference(List<Team> teams, Team selectedTeam) – метод, который вычисляет разницу между максимальным количеством очков в лиге и количеством очков выбранной команды.

* teams (List<Team>) - список всех команд
* selectedTeam (Team) - команда, для которой вычисляется отставание

Работа по следующему принципу:

* Находит максимальное количество очков среди всех команд (используя метод GetMaxPoints)
* Вычисляет разницу между этим максимумом и очками выбранной команды
* Возвращает полученную разницу

Void PrintPointsDifference(List<Team> teams, Team selectedTeam) – метод, который выводит в консоль информацию о том, на сколько очков выбранная команда отстает от лидера.

* teams (List<Team>) - список всех команд
* selectedTeam (Team) - команда, для которой выводится информация

Работа по следующему принципу:

* Вычисляет разницу в очках, используя метод CalcPointsDifference
* Форматирует и выводит сообщение в консоль