



Projekt programowanie obiektowe – Statystyki meczu koszykówki

KLASA „Stats”

pola klasy:

int pts, rbs, ast, fls, blks, stls, tos, fgm, fga, tpm, tpa, ftm, fta – zmienne na poszczególne statystyki.
Wszystkie chronione.

metody klasy:

Gettery dla wszystkich pól.

Trzy metody obliczające skuteczność procentową rzutów z gry, za 3 i osobistych

Metody *freeThrowTaken()*, *twoPointShotTaken()* oraz *threePointShotTaken* – które pytają o to, czy rzut został trafiony i w zależności od odpowiedzi właściwie modyfikują statystyki

Metody inkrementujące zmienne *rbs, ast, fls, blks, stls, tos*.

KLASA „TeamStats”

dziedziczy po Stats

pola klasy:

bool possession – czy drużyna ma posiadanie piłki

metody klasy:

bool getPossession – getter

void gainPossession – drużyna zyskuje posiadanie

void losePossession – drużyna traci posiadanie

void twoPointShotMade(), *threePointShotMade()*, *freeThrowMade()* -zmieniają punkty drużyny.

Używane gdy rzut został trafiony

void resetFouls() - ustawia faule drużyny na 0. Używana z końcem kwarty.

KLASA „Player”

pola klasy:

string name – imię zawodnika

string surname – nazwisko zawodnika

string number – numer zawodnika

string position – pozycja zawodnika

bool isActive - „true” jeśli jest na parkiecie, „false” wpp.

Stats st – statystyki zawodnika

Wszystkie pola tej klasy są chronione.

metody klasy:

void setPlayersData(string data[]) - metoda używana przez klasę Database. Ustala dane zawodnika przyjmując za argument tablicę napisów, które interpretuje kolejno jako numer, imię, nazwisko, pozycję, przynależność drużynową zawodnika. Ponadto ustawia pole *isActive* domyślnie na wartość „false”

void showPlayersData() - metoda wyświetlająca dane zawodnika

string getName() - getter

string getSurname() - getter

KLASA „Team”

dziedziczy po klasie Player

pola klasy:

string name – nazwa drużyny

vector <Player> players – zawodnicy tej drużyny

Player onCourt[5] – tablica zawodników aktualnie przebywających na parkiecie

TeamStats tst – statystyki drużyny

Wszystkie pola tej klasy są chronione.

metody klasy:

string getTeamName() - getter

void setName(string n) – setter

void showRoster() - wyświetla cały skład, korzystając z *showPlayersData()*

void setPlayers(Player plr) – setter

int getSize() - getter, zwraca ilość zawodników drużyny

KLASA „Database”

dziedziczy po klasie Team

pola klasy:

vector <Team> teams – zbiór drużyn w bazie danych

metody klasy:

void downloadData() - metoda pobierająca dane z pliku tekstowego o nazwie „teams.txt”. Dane w owym pliku muszą być wprowadzone w prawidłowy sposób, tj:

<NAZWA DRUŻYNY>

<NR> <IMIE> <NAZWISKO> <POZYCJA>

<NR> <IMIE> <NAZWISKO> <POZYCJA>

.

.

.

<NR> <IMIE> <NAZWISKO> <POZYCJA>

<NAZWA KOLEJNEJ DRUŻYNY>

itd.

Ilość zawodników w drużynie ograniczona jest do 12. Ilość drużyn natomiast, do 1000.

Metoda odczytuje kolejne linie pliku tekstowego. Pierwszą interpretuje jako nazwę drużyny, kolejne analizuje w następujący sposób. W tablicy stringów *words* przechowywać będzie docelowo dane nt. jednego zawodnika. Funkcja odczytuje całą linię. A następnie dodaje znak po znaku do pierwszego elementu tablicy *words*, dopóki nie napotka znaku spacji lub końca linii. W ten sposób uzupełnia tablicę danymi nt. zawodnika, by następnie przekazać je jako argument funkcji *setPlayersData* dla obiektu *p* typu *Player* utworzonego na starcie działania funkcji. W kolejnym kroku wywołuje metodę *setPlayers* dla obiektu *Team t*, przekazując jako argument obiekt *p*, któremu przed chwilą nadano odpowiednie wartości.

void showTeams() - metoda używająca *showRoster()*, by wyświetlić wszystkie drużyny

KLASA „Game”

pola klasy:

Team teamA, teamB – rywalizujące drużyny

int quarter – zmienna przechowująca informacje nt. kwarty

metody klasy:

Game() - konstruktor, ustawia quarter na 1

void selectTeams() - metoda służąca do wyboru drużyn. Wywołuje kolejną metodę
void selectStartingFive() -metoda służąca do wyboru wyjściowej piątki obu drużyn. Wywołuje kolejną metodę
void gameplay() - metoda służąca do właściwej rozgrywki. Wyświetla pełne statystyki i oprócz tego wywołuje kolejną metodę z argumentem typu string podanym przez użytkownika
void whatToDo(string action) – metoda, która w zależności od wpisanej przez użytkownika wymaganej akcji modyfikuje statystyki poszczególnych graczy