

PROGRAMAZIO MODULARRA ETA OBJEKTUEI BIDERATUTAKOA

AZTERKETA PARTZIALA –2015ko Apirilak 1 – B Bertsioa

Izen Abizena: _____

Ligako Futbol Simuladorea (LFS)

Ligako futbola simulatzeko aplikazio bat garatu nahi dugu. Horretarako, ligan parte hartzen duten klub guztien zerrenda bat izango dugu. Klub bakoitzeko, (ariketa honen garapenerako at gelditzen diren beste datu batzuen artean) bere izena (*String*) eta kontratatuta dituen jokalarien zerrenda gordeko da (aurrerago jokalaria hauen zehetasun gehiago emango dira).

Aipatutako informazioz gain, egun arte jokatu diren partiduen emaitzak ere gordeko dira. Emaitza bakoitza lau atributuz adieraziko da (informazio gehigarria ariketa honen eremutik kanpo gelditzen da): etxeko kluba, kanpotarra, etxekoaren gol zerrenda, eta kanpotarrarena. Gauzak horrela, golen zerrendan, gol bakoitzeko, gola sartu duen jokalaria informazioa izan behar dugu (*Jokalaria* klasekoa) eta gola zein minutuan gertatu den (*int*).

Partidu bat denbora errealean simulatzea ez da ariketa honen helburua, baizik eta partidu baten zati batzuk simulatzea soilik. Jarraian, erabiliko diren hainbat klase aipatuko ditugu:

Baloia EMA: **Hau jada inplementatuta dagoela suposatu behar duzue, beraz zuek ez duzue ezer programatu behar klase honen barruan.** Bertan dauden metodoak erabili baino ez duzue egin behar. Erabili beharko duzuen metodoaren burua hurrengoa da:

- *public void desplazatu(double pIndarra, double pAngelua)*. Metodo honi deitzearen emaitza gisa baloia zelaiaren puntu batetik bestera mugituko da, pasatzen zaion indarra eta angeluaren arabera. Baina berriro, mugimendu hau nola gauzatzen den zuentzat ez du garrantzirik izan behar.

Klub bakoitzaren Jokalariei dagokionez, jokalaria mota desberdinak daude: atezainak, defentsak, erdilariak, eta aurrelariak. Jokalaria guztiek, izena (*String*), dorsal zenbakia (*int*), eta baloia jaurtitzean eman diezaioketen indar maximoa (*double*) izango dute ezaugarri gisa.

Jokalariek, oro har dituzten metodoen artean hurrengoak dira aipagarri eta guretzat adierazgarri:

- *private double teklatutikTartekoBalioaEskatu(double pBeheLim, double pGoiLim)*. Metodo hau jada programatuta dago, beraz ez da inplementatu behar. Metodo honi, *indarraHautatu()* eta *angeluaHautatu()* metodoek dei egingo diote teklatutik balio bat irakurri ahal izateko, hau da, jokalaria hautatu dezan ze nolako indarra eta angeluarekin baloi jaurti nahi duen. Metodo honek, teklatutik balio bat irakurtzeaz gain, balio hori pasatuko zaizkion *pBeheLim* eta *pGoiLim* artean egotea konprobatuko du.
- *protected double indarraHautatu()*. Metodo honek aipatu berri dugun *teklatutikTartekoBalioEskatu()* metodoari deituko dio, 0.1 eta jokalaria duen indar maximoaren arteko balio bat bueltatu diezaion.
- *protected double angeluaHautatu()*. Metodo honek, aurrekoak bezala, *teklatutikTartekoBalioEskatu()* metodoari deituko dio, 0.0 eta 359.9 dagoen balio bat bueltatu diezaion.
- *protected void jaurti()*. Metodo honek Baloiaren *desplazatu()* metodoari deituko dio, (*Baloi.getNireBaloi().desplazatu(indarra, angelua)*). Defektuz, eta kontrakoa adierazi ezean, indarra, *indarraHautatu()* metodoak bueltatutakoa izango da, eta angelua *angeluaHautatu()* metodoak bueltatzen duena, hurrenez hurren. Oharra: Aurrerago xehetasun gehiago emango dira.

Atezainak, baloia eskuekin hartu/ukitu dezaketen jokalaria bakarrik dira. Hori dela eta, *public void baloiaEskuekinHarrapatu()* metodoa izango dute. Metodo hau ez da inplementatu behar baina klase diagraman behar den lekuan ipini beharko duzue.

Kontsideratuko dugu, partidua aurrera joan ahala, atezainak apenas mugitzen direnez, ez direla nekatzen, ez aldiz gainontzeko jokalaria, hauek bai nekatzen direla. Hori dela eta, zelaiko jokalaria (hau da atezainak ez direnak) *nekea (int)* atributuz ere hornituta egongo dira. Atributu honen balioa eguneratzen joango da, jokalaria zelaatik zehar mugitzen denean, baina honen kudeaketa ariketa honen eremutik kanpo gelditzen da. Bakarrik jakin behar da, jokalaria bakoitzaren atributu honen balioa eguneratzen joango dela automatikoki.

Zelaiko jokalariek ere, metodo berezi bat dute (berriro, atezainek ez dutena) *public void baloiaPaparrarekinKontrolatu*. Metodo hau ez da inplementatu behar.

Baloia jaurtitzeko garaian (*jaurti()*) metodoa erabiliko da. Atezainak nekatzen ez direnez, beraiei ez die eraginik izango *Nekea* atributuak, bai ordea zelaiko jokalariei. Jaurtiketa, baloiari eman nahi dioten indarra eta angeluaren arabera egingo dute. Baina *nekeak*, *indarraHautatu()* eta *angeluaHautatu()* metodoei deitzean lortutako balioengan eragina izango du, balio hauek apur bat aldatuz, errore bat sartuz, jarraian azalduko denez. Hartu adibidez defentsa bat. Berak erabaki dezake baloia jaurtitzea 10eko indarra eta 90ko angeluarekin baina 5eko nekea duenez, errealitatean balioa 9.5eko indarra eta 90.5eko angeluarekin *jaurtiko()* du. Ikusi xehetasunak:

Defentsentzat, *nekeak* atributua n balioa balu, orduan:

- Errealitatean balio jaurtitzeko indarraren balioa *indarraHautatu()* minus n izango da.
- Angeluari dagokionez, bere balio kalkulatzeko n bikoitia balitz *angeluaHautatu()* minus $n/10$ izango da. Aldiz, n bakoitia balitz minus izan beharrean gehi izango da.

Erdilariantzat, *nekeak* atributua n balioa balu, orduan:

- $n < 20$, errealitatean baloia jaurtitzeko indarraren balioa *indarraHautatu()* minus $(0.9*n)$ izango da. Aldiz, $n \geq 20$ balitz, indarraren balioa *indarraHautatu()* minus $(1.2*n)$ izango da.
- Angeluari dagokionez, bere balioa *angeluaHautatu()* metodoak bueltatutakoa izango da, ez da nekeagatik modifikatuko.

Aurrelariak, gainontzeko jokalariek ez duten beste atributu bat daukate: *Zortea (int)*. Aurrelariantzat, baloia jaurtitzeko momentuan nekeak n balioko balu, eta zorte atributuak z , orduan hurrengoa bete beharko da:

- $z \geq n$, baloia jaurtitzeko indarra *indarraHautatu()* metodoak bueltatzen duena izango da, aldaketarik gabe. $z < n$, balitz indarra minus $(n - z)$ izango da.
- Angelua, *angeluaHautatu()* metodoak bueltatutakoa izango da.

Eskatzen dena:

- **(XXX puntu)** Klase diagrama marraztu, adieraziz klase bakoitzeko zeintzuk diren atributuak eta metodoak. Eta zeintzuk diren ere, klaseen arteko erlazioak. Diseinuak, informazio partekaketa maximizatzea lortu behar du.
- **(XXX puntu)** *EmaitzaZerrenda* klaseari dagokion *public int bakartiKopurua(Jokalaria pJokalaria)*. Metodo honek, gaur egun arte *pJokalaria* gol-sartzaile bakarra izandako partiduen kopurua

bueltatuko du. Oharra: jokalaria bati *bakarti* bat kontabilizatzen zaio baldin eta partidu batetan bere taldeko gol-sartzaile bakarra izan denean eta behintzat gol bat sartu duenean.

- (XXX puntu) Jokalaria eta bere azpi-klase guztien eraikitzaileak inplementatu, baita dauden EMA klaseenak ere.
- (XXX puntu) *indarraHautatu()*, *angeluaHautatu()* eta *jaurti()* metodoak inplementatu beharrezkoa den klase guztietan, argi utziz ze klaseari dagokion inplementazio bakoitza.

OHARRAK:

1 - Ez da aurreko ataletan eskatzen ez den ezer inplementatu behar.

2 - Penalizazioa jasoko duzue baldien eta:

- Beharrezkoak ez diren metodoen inplementazioak, nahiz eta beste testuinguru batean beharrezkoak izan.
- *Private* izan behar diren metodoak *public* jartzea.
- Orain arte ikusi eta erabilitako arauak ez jarraitzea, alegia, elementu desberdinen izenak hautatzeko arauak (atributuak, metodoak, parametroak, aldagaiak).