Jaizveido C# virtualais robots:

**Robots** sastāv no 2 klasēm: (***piemēram***):

**iRobot** -galvenā mašīna

**ProximitySensor** – kas būs IRobota “acis” (divi objekti)

\*\*

Klase **ProximitySensor**, kas sastāv no

----***laukiem*** (properties):

public state (on/off)

public serial\_number

----***metodēm (funkcijām)***:

Klase **ProximitySensor** satur *2* ***konstruktors****:*

Pirmais – piešķir RANDOM sērijas numuru un izslēgto (**off**) mainīgam *state*,

Otrais-piešķir **parametrā** doto sērijas numuru un izslēgto (**off**) mainīgam *state*,

Get/Set nodrošina **ievadīšanas aizliegumu**, bet **lasīšanas** atļauju *state* mainīgam

Public bool Scan(“area”) – atgriež vai priekšā ir šķērslis, (“area” ir **pāra**(ir)/**nepāra**(nav) )cipars

public void PrintSensor – izvada informāciju par sensoru

\*\*

Klase **iRobot**, kas sastāv no

-----***laukiem*** (properties):

public model\_name

pprivate LeftSenor (objekts no **ProximitySensor)**

private RightSensor

private alarm[] (vērtības **red**/**green**)

private string map[] – kas sastāvēs no simboliem ‘**r’**(right), ‘**l**’(left) un ‘**f**’(front) atkarībā no virziena

----***metodēm (funkcijām)***:

Klase **iRobot** satur *2* ***konstruktors****:*

Pirmais – piešķir RANDOM sērijas *model\_name*,

Otrais-piešķir **parametrā** doto sērijas modeli *model\_name*,

public void SwitchOnOff – ieslēdz robotu, ieslēdzot abus sensorus

public void Run(“room”) - “room[]” ir int[10] masīvs ar skaitļiem 10-99. Aizpilda masīvus map[] un alarm[] lasot pa elementam no room[] masīva.

Private void PrintRobot – izvada robota maršrutu map[] un alarm[] statusu no Run metodes.

**Procesa apraksts.**

Robotu var palaist “Run” tikai ja tas sākumā ir ieslēgts “SwitchOnOff”.

Metode “Run” saņem kā parametru iepriekš sagatavoto int[10] masīvu ar skaitļiem 10-99. Pirmais cipars nodrošina informāciju kreisajam sensoram, otrais cipars labajam. Secīgi apstrādājot masīvu notiek map aizpildīšana ar ‘r’,‘l’,‘f’ simboliem atkarībā no informācijas no sensoriem.

Ja robots SPĒJ iztīrīt visu maršrutu map veidojas secība no 10 virzieniem, un alarm saņem STATUSS=>green.

Ja robots ir STRUPCEĻĀ (abi cipari skaitlī ir pāra) – robot apstājas, nepabeidzot tīrīšanu neaizpildot līdz galam map, un saņem STATUSS=>red.

\*\*

No bāzes klases ( satur Main metodi)

1)tiek izsaukti divi TESTA roboti,

vienam jāiedod maršrutu, bez strupceļa, otrajam ar.

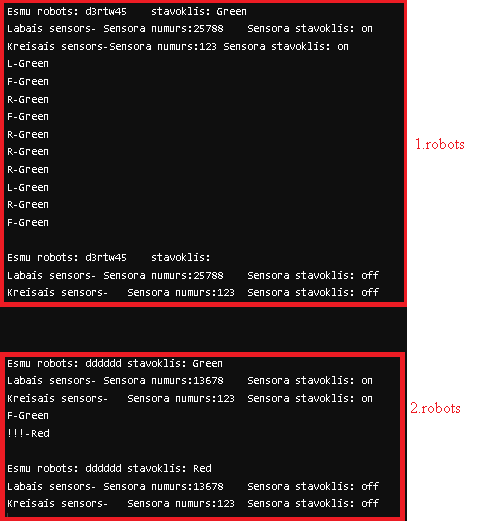
2)Jāieslēdz roboti,

3)iedarbina robotus

Izdrukā rezultātu

4) izslēdz robotu

**Rezultāta piemērs.**



**Skelets.**

using System;

namespace Robots {

**public class Program {**

**public static void Main() {**

int[] ????? = {???????}; // telpas masīvs 10 skaitli no 10 lidz 99

int????? = {???????}; // telpas masīvs 10 skaitli no 10 lidz 99

IRobot Irobot1 = new IRobot();

Irobot1.SwitchOnOff(); // ieslēdz robotu

Irobot1.Run(telpa1); // darbina robotu

Irobot1.SwitchOnOff(); // izslēdz robotu

Console.Write("\n\n\n");

IRobot Irobot2 = new IRobot("dddddd");

Irobot2.SwitchOnOff(); // ieslēdz robotu

Irobot2.Run(telpa2); // darbina robotu

Irobot2.SwitchOnOff(); // izslēdz robotu

**}**

**}**

**class IRobot {**

private ProximitySensor ???? = new ProximitySensor(123);

// pieliek objektu ProximitySensor ar parametru

private ProximitySensor ???? = new ProximitySensor();

// pieliek objektu ProximitySensor bez parametra

private static string[] ???? = new string[10]; // izdrukas masīvs VIRZIENIEM: burti L,R,F

private static string alarm=""; // robota STĀVOKLIS:teksts Green,Red

public string model\_name{

get{????;} // atgriež konstuktora piesķitro vērtību

}

**public IRobot() {** // konstruktors bez parametra

???? // noģenerē modeli

**}**

**public IRobot(string ????) {** // konstruktors ar parametru

????? // piešķir modeli

**}**

**public void SwitchOnOff() {**

???? // parbauda stāvokli

Console.Write("Labais sensors-");

????.PrintSensor();

Console.Write("Kreisais sensors-");

????.PrintSensor();

**}**

**public void Run(int[] ????) {**

???? // pāarbaudes

for (????){// cikls pa saņemto masīvu

???? // pārbauda sensora rezultā un aizpilda virzienu masīvu un alarm

}

PrintRobot();

**}**

**static void PrintRobot() {**

for (???) {// cikls pa virzienu masīvu

???//izdruka virzienu un alarm stāvokli

}

**}**

**}**

**class ProximitySensor {**

**public ProximitySensor() {**// konstruktors bez parametra

??? // noģenerē serialo numuru

**}**

**public ProximitySensor(int ser) {**// konstruktors ar parametrua

??? // piešķir serialo numuru

**}**

**public string state {**

set {??? } // ievadīta stāvokļa pārbaude un piešķiršana

get {???} // stāvokļa atgriešana

**}**

**public int serial\_number {**

get {???}// atgriež serialo numuru

**}**

**public void PrintSensor() {**

??? // izdrukā sensora stavokli un numuru

**}**

**public bool Scan(???) {**

??? // parbauda vai iegutais skaitlis ir PĀRA

**}**

**}**

}

**Virzienu zurnals:**

R, F, F,.....

**SwitchOnOff**

Alarm:  
Green/Red

**Telpa:**

10,21,31.......

True/false

True/false