Elektronski Fakultet Niš Djordje Lazić 17722

Mladen Antić 17545

Arhitektura i projektovanje softvera

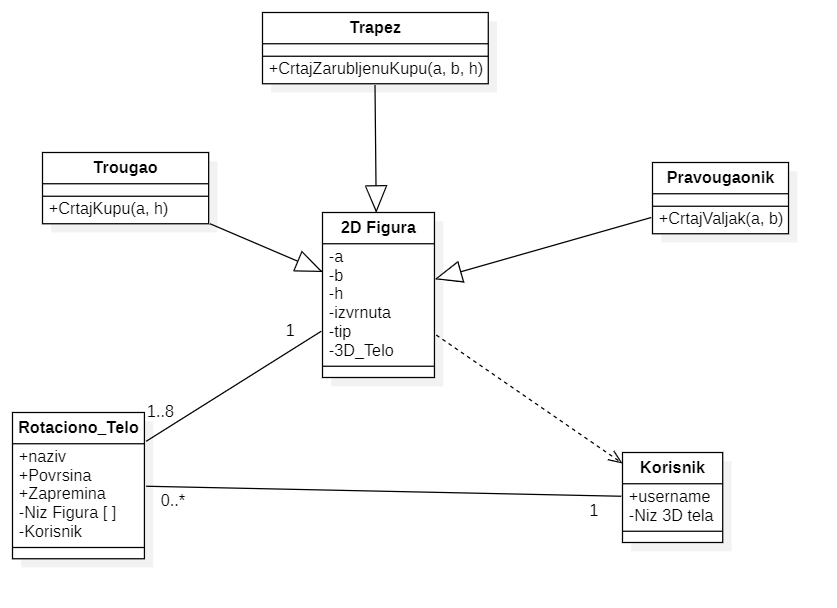
Projekat faza 2

Business Model:

Osnovna geometrijska tela u vidu trapeza , pravougaonika i trougla su izvedena iz klase 2D Figura , a ujedno i njihovi parametri. Svaka klasa ima funkciju za sopstveno iscrtavanje odgovarajućeg 3D tela. Parametri a,b,h predstavljaju dužinu , širinu i visinu , dok promenljiva "izvrnuta" služi da odredi da li se telo koje je na dnu ili na vrhu rotacionog tela koristi za "izdubljivanje" tela iznad ili ispod njega ( pravimo šupljinu umesto da dodamo telo na dno odnosno vrh 3D tela ). Tip figure kaže da li se radi o trouglu , pravougaoniku ili trapezu.

2D figura ima pokazatelj na 3D figuru kojoj pripada i predstavlja osnovu za istu.

3D figura se dobija rotacijom 2D tela oko svoje ose. Rotaciono telo predstavlja 3D figuru i ona ima svoj naziv i izračunatu poršinu i zapreminu na osnovu unetih parametara. Niz figura je skup tela koji su uneti u viewport i za koje se računa površina i/ili zapremina . Svaka rotaciona figura ima svog jedinstvenog korisnika koji ju je napravio. Korisnik ima svoj username koji sam bira i ima uvid u sva tela koja je prethodno uneo u viewport.



Model Perzistencije:

Figure.js:

const mongoose = require('mongoose');

const figureSchema = new mongoose.Schema({

  a:Number,

  b:Number,

  h:Number,

  figura:String,

  username:String

},{ versionKey: false });

const FigureModel = mongoose.model("figure", figureSchema);

module.exports = FigureModel;

3DTelo.js:

const mongoose = require('mongoose');

const rotTeloSchema = new mongoose.Schema({

  naziv:String,

  P:Number,

  V:Number,

  nizFigura:Array,

  uID:String

},{ versionKey: false });

const rotTeloModel = mongoose.model("figure", rotTeloSchema);

module.exports = rotTeloModel;

user.js:

const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

  username:String,

  myProjects:Array

},{ versionKey: false });

const userModel = mongoose.model("figure", userSchema);

module.exports = userModel;

Ove komponente se eksportuju u glavni server (server.js) I koriste se za mapiranje podataka u MongoDB bazu, I sa njihovim parametrima se vrši obrada podatka.

Za mapiranje u bazu je korišćen Repository obrazac:

FigureRepository.js:

const FigureModel = require('./figure');

class FigureRepository {

  constructor() {}

  async create(figureData) {

    try {

      const figure = await FigureModel.create(figureData);

      return figure;

    } catch (error) {

      throw error;

    }

  }

  async findById(id) {

    try {

      const figure = await FigureModel.findById(id);

      return figure;

    } catch (error) {

      throw error;

    }

  }

  async findAll() {

    try {

      const figures = await FigureModel.find();

      return figures;

    } catch (error) {

      throw error;

    }

  }

  async update(id, updateData) {

    try {

      const figure = await FigureModel.findByIdAndUpdate(id, updateData, {

        new: true,

      });

      return figure;

    } catch (error) {

      throw error;

    }

  }

  async delete(id) {

    try {

      const result = await FigureModel.findByIdAndDelete(id);

      return result;

    } catch (error) {

      throw error;

    }

  }

}

module.exports = new FigureRepository();