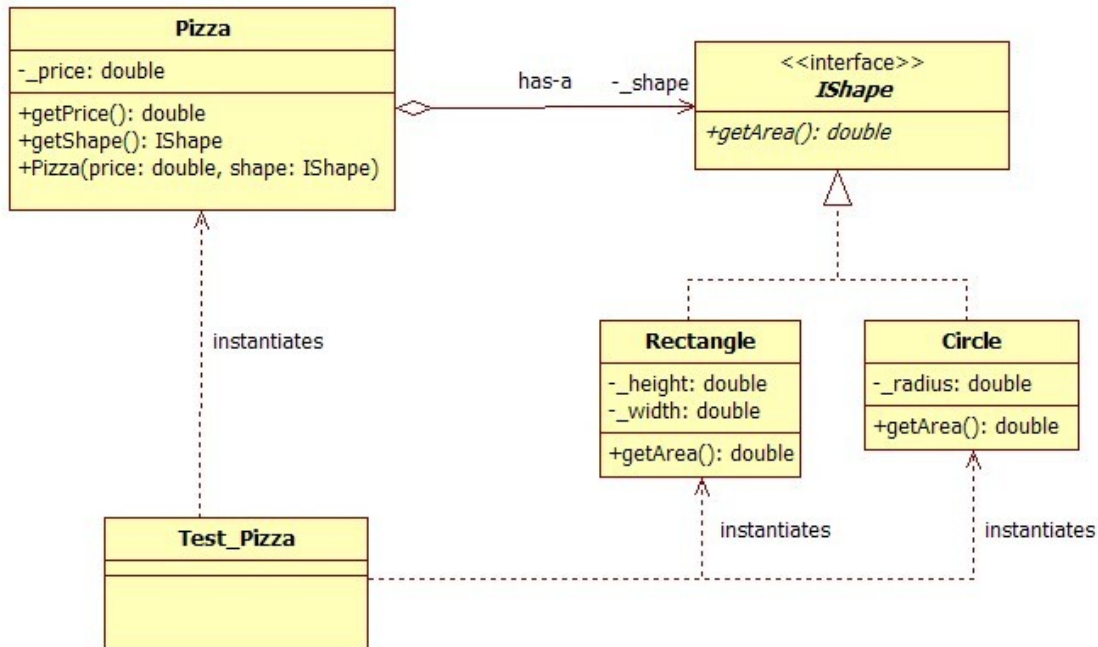


STAR UML - alat za modelovanje UML dijagrama

STAR UML je alat za kreiranje raznih vrsti dijagrama u UML-U. On omogućava automatsko generisanje JAVA koda na osnovu dijagrama I obrnuto, generisanje dijagrama u skladu sa određenim JAVA kodom.

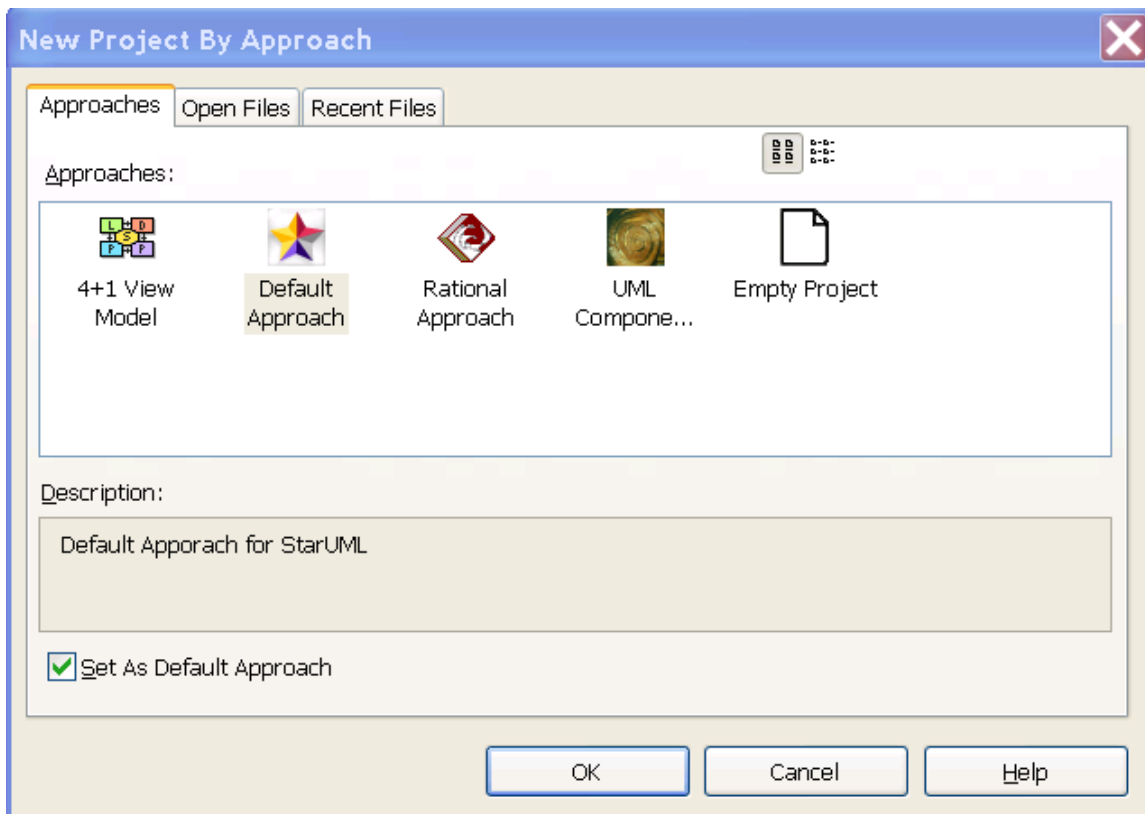
Primer za kreiranje:



Kada je STAR UML instaliran , startujte program dvostrukim klikom miše na ikonicu.

1. korak

Nakon startovanja programa, trebalo bi da se pojavi templejt stranica **"New Project by Approach"** koja ima izgled kao na slici ispod. Ukoliko je tako, izaberite **"EMPTY PROJECT "** i kliknite **OK**.



2. korak

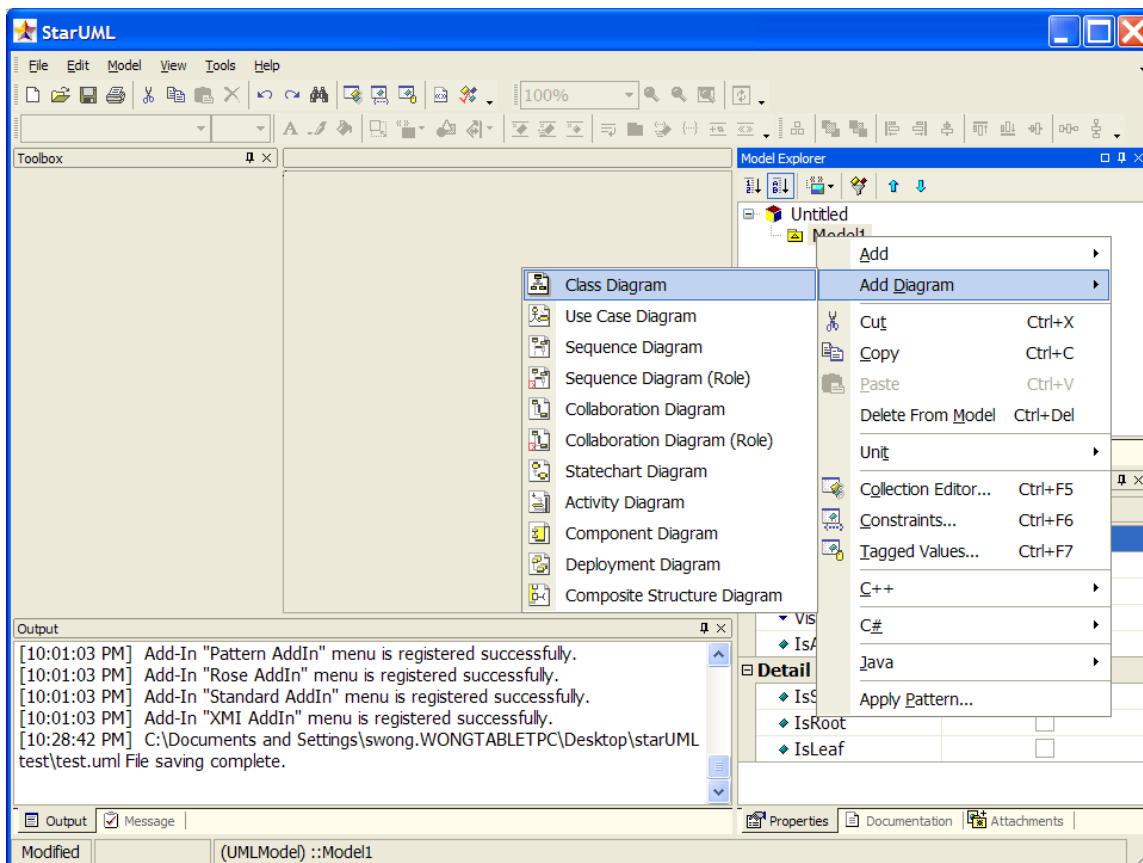
U **MODEL EXPLORERU** sa desne strane izaberite **Untitled**

3. korak

Na glavnom meniju, izaberite **MODEL** i opciju **ADD** i izaberite **MODEL**.

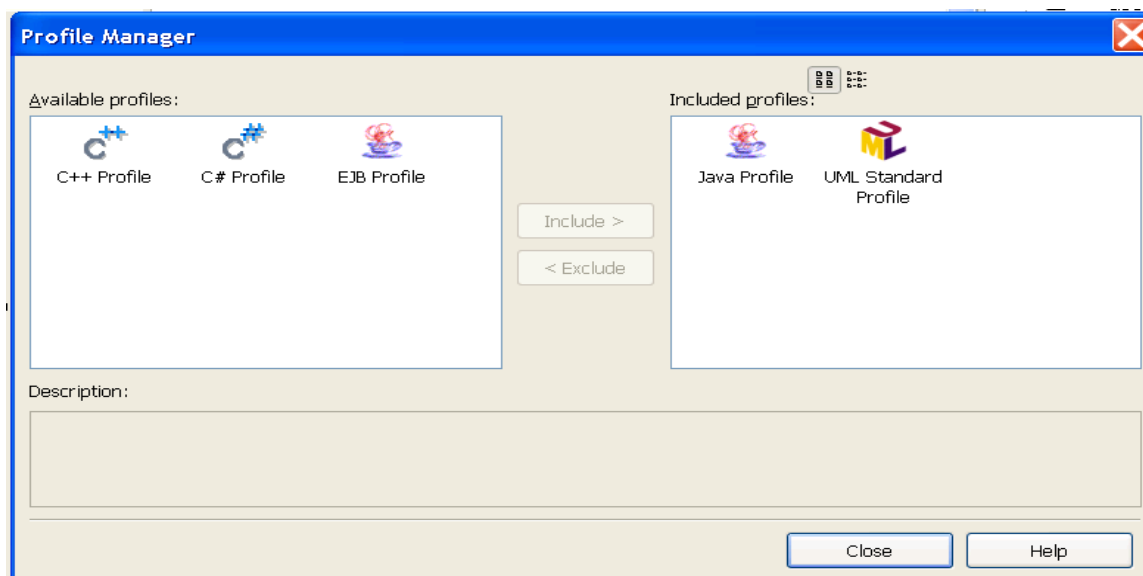
4. korak

Kao na slici ispod, u **MODEL EXPLORER** panelu izaberite opciju **ADD DIAGRAM**, pa nakon toga **CLASS DIAGRAM**.



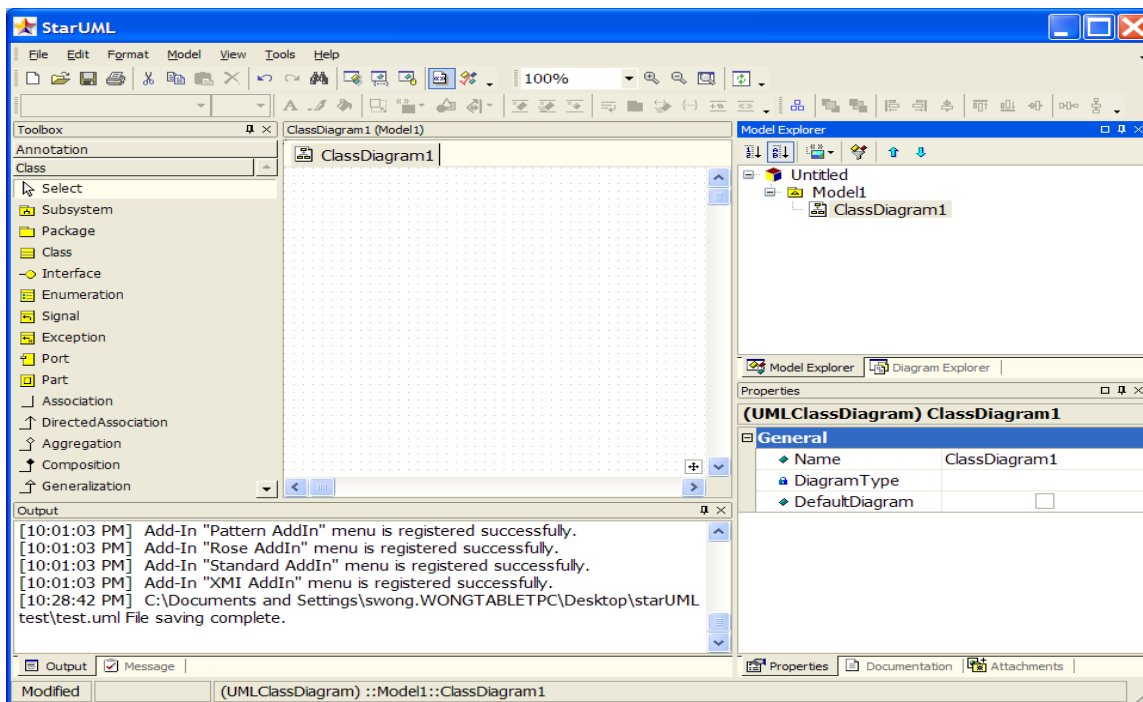
5. korak

Idite na **MODEL PROFILE** u okviru **MODEL EXPLORERA**, da biste podesili profil modela koji ćete koristiti. Budite sigurni da ste uključili **JAVA PROFILE**. Grafički prikaz je na slici ispod.



6. korak

Sačuvajte projekat odmah da ne bi došlo do neželjenih situacija. Izaberite **FILE, SAVE AS** i odaberite ime i lokaciju za vaš projekat. Nakon toga, vaš projekat bi trebalo da ima izgled kao na slici ispod.



7. korak

Sada trebate početi stvarni rad sa dijagramom koristeći **TOOLBOX**, pozicioniran na levoj strani ekrana. Izaberite **class** i kliknite levim klikom miša bilo gde u prozoru. Ovo bi trebalo da stvori novu klasu sa generisanim programskim imenom. Promenite joj ime u **circle**.

8. korak

Dodajte atribut (ili polje) kreiraniom **circle-u**, desnim klikom objekta na dijagramu, biranjem **ADD MENU**, pa **Attribute** opcije. Ukucajte za **name _radius**.

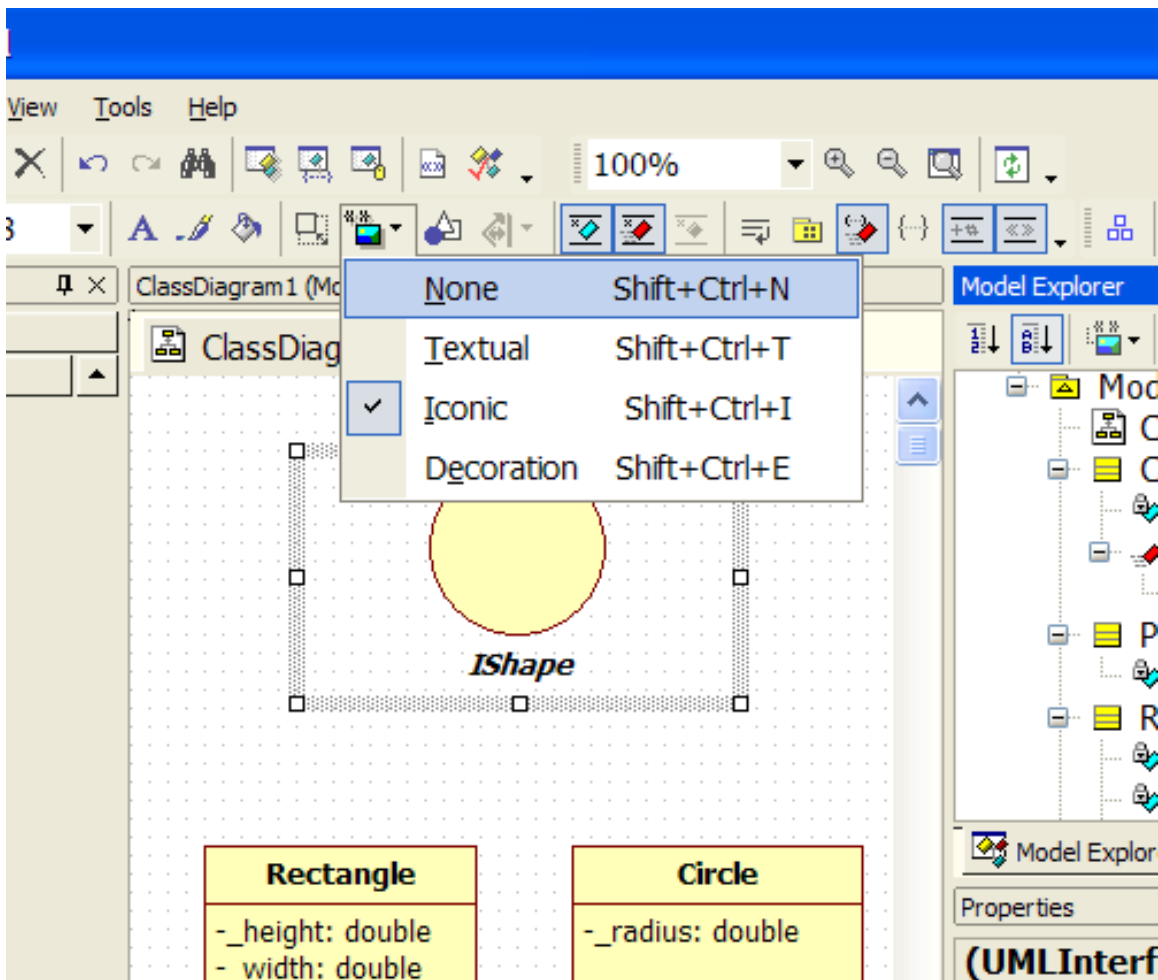
- Specificirajte tip podatka u **Properties** panelu da bude **double**
- Visibility atributa podesite na **private**. Ako ima potrebe možete je promeniti.

9. korak

Ponovite isti postupak pri kreiranju klasa **RECTANGLE** sa privatnim atributima **_height** i **_width**.

10. korak

- Kreirajte interfejs **ISHAPE**. Iz **toolbox-a** izaberite opciju **Interface** i kliknite negde na paletu. Promenite programski generisano ime na **ISHAPE**.
- U toolbaru, desni klik na Ishape, pa odaberite **Format** i izaberite **Stereotype Display** i promenite vrednost na **none**. Ovo će promeniti prethodni cirkularni oblik u pravougaoni oblik.
- Isto, u toolbaru pri vrhu, deselektujte **Suppress Operations** opciju.



Dodajte *getArea* metod tipa *double* u okviru *ISHAPE* interfejsa.

- Ovo se postiže desnim klikom na interfejs biranjem opcije *Add* i klikom na crveno *Operation* dugme. Ime je kao što je već rečeno *getArea*. Cekirajte opciju *IsAbstract*.
- Da biste odabrali (povratni) tip metode, u *MODEL EXPLORERU*, desni klik na metod koji smo upravo kreirali, izaberimo *Add Parameter*, pa u *Properties box* za ime ništa ne menjamo, za *Direction kind* izabere *return*, a za tip *double*.

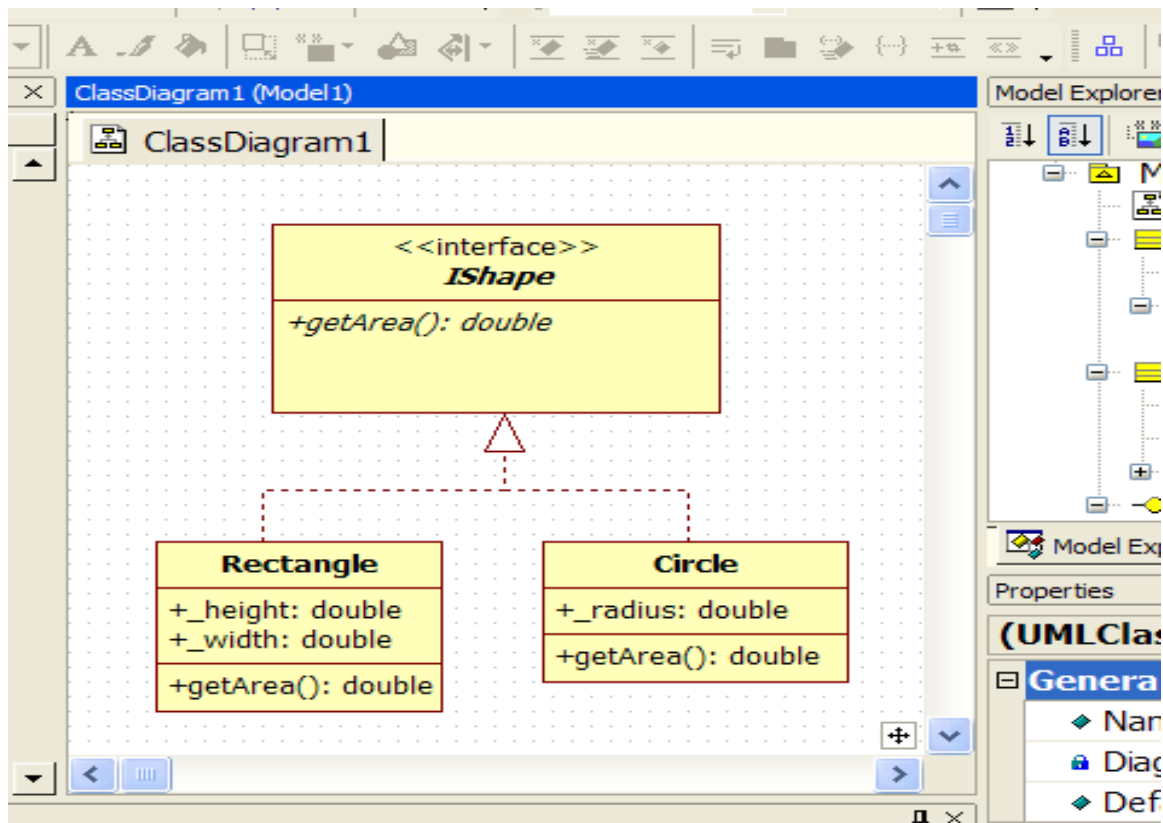
11. korak

Da bi *rectangle* i *circle* implementirali interfejs *ISHAPE*, izaberite *realization* strelicu iz *toolboxa* i povežite *rectangle* i *circle* sa *ISHAPE*.

12. korak

Kako *circle* i *rectangle* implementiraju *ISHAPE* interfejs, obe moraju imati isto ponašanje (metode) kao taj interfejs.

- U *MODEL EXPLORER TOOLU* iskopirajte *getArea* metod iz *ISHAPE* u *circle* i *rectangle*
- Ove metode u *circle* i *rectangle* nisu apstraktne, ali logički gledano izvršavaju istu akciju (računaju površinu) pa se treba unčekirati opcija *IsAbstract*



13. korak

Vaš dijagram treba da izgleda kao na slici iznad

14. korak

Dodajte klasu *Pizza*

- Privatan atribut *_price* tipa *double*
- Javna metoda *getPrice* tipa *double*

15. korak

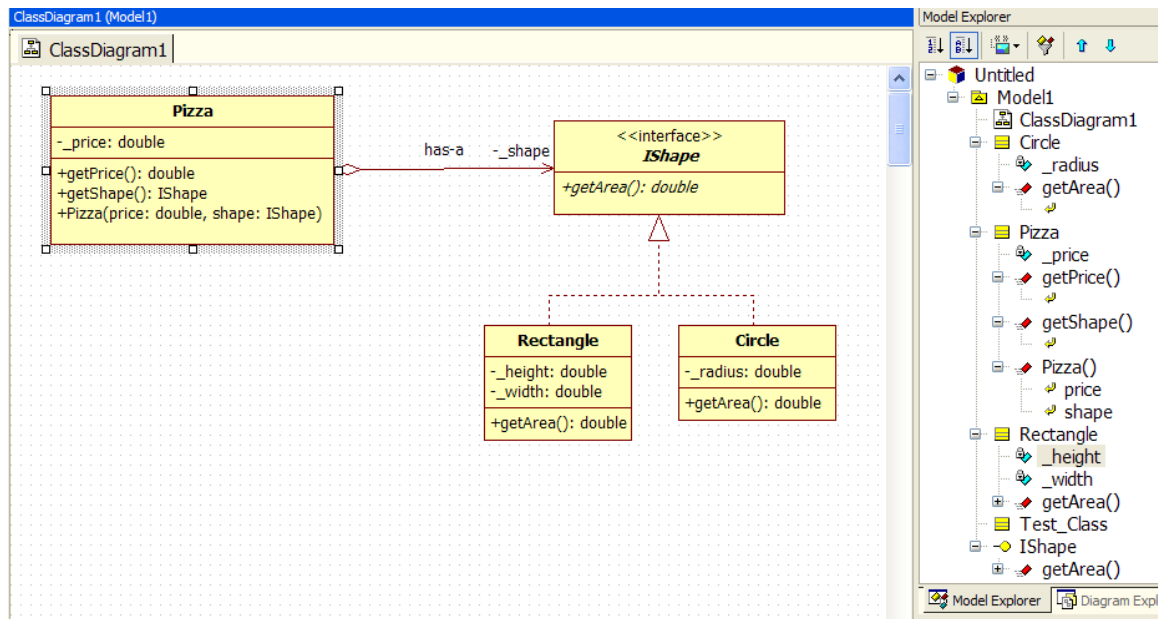
Da bi *Pizza* imala referencu sa *ISHAPE*, izaberite klasu *Pizza*

- Izaberite vezu *DirectedAssociation* iz toolboxa i od *pizza* ka *ISHAPE* izvršite povezivanje
- Nakon povezivanja, kliknite na strelicu i promenite joj ime sa *End1.Aggregation* na *AGGREGATE*

16. korak

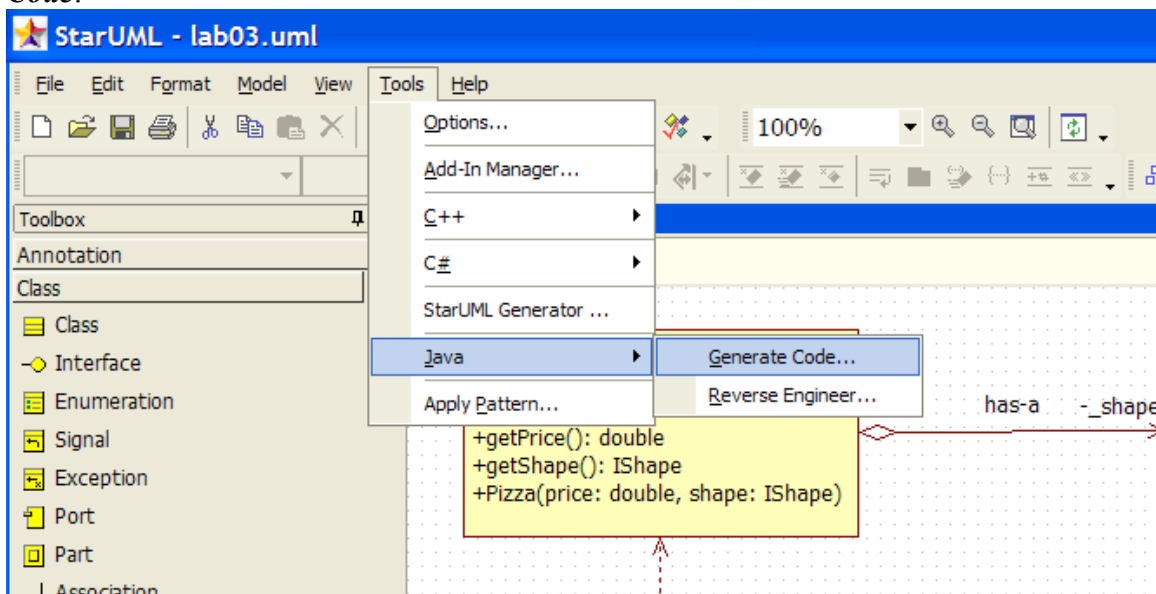
Konstruktori su posebni delovi koda kada želimo da inicijalizujemo instance klase

- Da bi dodali konstruktor za *Pizza*, uradite desni klik na *Pizza* i iz *Add* menija izaberite *Operation*. Nazovite metod *Pizza* i dodajte parametre, tipa *double* za *price* i *ISHAPE* za *shape*
- Dodajte *circle* konstruktor sa parametrom *double radius*
- Dodajte *rectangle* konstruktor sa parametrima int *height* i int *width*



17. korak

Da biste generisali JAVA kod , idite na **tools** na meniju pri vrhu, izaberite **JAVA** , pa **Generate Code**.



- Iz **dialog box-a**, izaberite model, verovatno nazvan **modell** i idite na **next**
- Izaberite opciju **select all** i idite na **next**
- Izaberite zeljeni **output** direktorijum i idite na **next**
- U **Option setup** obavezno cekirajte **Generate the Documentation by JavaDoc** i **Generate empty JavaDoc**. Ostale opcije ne treba cekirati
- STAR UML ce nakon ovog generisati kod na osnovu nasih dijagrama. Za izlazak idite na **finish**
- Mozete u skladu sa gore opisanim da dopunite kod da biste dodati zeljene i potrebne funkcionalnosti

Zadaci za samostalan rad

1. Nacrtati dijagram klasa na jeziku UML za sledece klase i generisati Java i C# kod:

Tacka u ravni se zadaje realnim koordinatam koje mogu da se dohvate. Moze da se izracuna

rastojanje do zadate tačke.

Apstraktna **figura** u ravni se zadaje tačkom koja predstavlja centar figure i koja može da se dohvati. Figura može da se premesti na novo mesto i da se pomeri za zadati pomak duž koordinatnih osa. Može da se izračuna obim i površina figure i da se odredi rastojanje od centra figure.

Krug u ravni je figura zadata poluprečnikom koji može da se dohvati.

Trougao u ravni je figura zadata dužinama stranica koje mogu da se dohvate.

2. Nacrtati dijagram klasa na jeziku UML za sledeće klase i generisati Java i C# kôd:

Predmet: ima jedinstven, automatski generisan broj ID koji može da se dohvati

Skladište: može da sadrži zadat broj predmeta. Stvara se prazan, posle čega predmeti mogu da se stavljaju i vade jedan po jedan. Predmeti se vade po redosledu stavljanja. Može da se dohvati broj predmeta u skladištu i da se ispita da li je skladište puno i da li je prazno.

Proizvođač: može da napravi jedan predmet i da ga stavi u skladište koje se zadaje prilikom stvaranja proizvođača

Potrošač: može da uzme jedan predmet iz skladišta koje se zadaje prilikom stvaranja skladišta.