

DOKUMENTACIJA

“Klase i interfejsi u mini-C” Sintaksna i semantička analiza

1. Osnovni podaci

Broj indeksa	Ime i prezime	Tema
SW68/2019	Mičo Milić	Klase i interfejsi u minic-u

Korišćeni alati

Naziv	Verzija
Flex	2.5.35
Bison	2.3
GCC compiler	Apple clang 12.0.5

2. Evidencija implementiranog dela

U projektu su odradjeni zadaci podržavanja klasa i interfejsa u mini-C jeziku po uzoru na java klase i interfejse. Za oba je uradjena leksička, sintaksna i semantička analiza. Interfejsi kao u javi predstavljaju spisak metoda, klase koje implementiraju taj interfejs moraju implementirati njegove metode.

3. Detalji implementacije

Implementacija se sastoji od deklarisanja klasa i interfejsa, deklarisanja klasa koje implementiraju interfejse, instanciranja objekata tih klasa i rukovanje njihovim metodama i atributima. Broj klasa u fajlu je proizvoljan, isto važi i za broj interfejsa. Primer deklaracije i upotrebe jedne klase prikazan je na Slici 1.

```

class Person{
    int age;
    int wage;
    int experience;

    Person() {}

    Person (int age, int wage, int experience){
        age = age;
        wage = wage;
        experience = experience;
    }
}

int main() {
    int age;
    int wage;
    int experience;
    age = 22;
    wage = 1000;
    experience = 1;

    Person p1 = new Person(age, wage, experience);
}

```

Slika 1 Primer definisanja klase "Person" i njeno instanciranje pomoću parametrizovanog konstruktora

Na slici se vidi primer definicije jedne klase "Person" koja ima parametrizovani konstruktor i konstruktor bez parametara. U main funkciji se vidi primer kreiranja jednog objekta te klase. Šablon za instanciranje je Klasa varijabla = new Klasa (argumenti_konstruktora...); Po uzoru na javu u klasama je podržan proizvoljan broj konstruktora sa proizvoljnim brojem parametara. Ukoliko se kreiraju 2 konstruktora sa istim brojem argumenata i istim tipovima dolazi do constructor duplicate greške. Da bi se podržalo deklarisanje više klasa sa više raznih konstruktora, funkcija tabela simbola je proširena sa dodatnim atributom parent_idx pomoću kog se pri analizi može pronaći roditelj svakog elementa.

Na Slici 2 je prikazan primer definisanja interfejsa i klase koja implementira taj interfejs.

```

interface Employee {
    int calculateVacationDays();
    int calculateYearsTillRetirement();
}

class Person implements Employee{
    int age;
    int wage;
    int experience;

    Person() {}

    Person (int age, int wage, int experience){
        age = age;
        wage = wage;
        experience = experience;
    }

    int calculateVacationDays(){
        int days;
        days = experience + 25;
        return days;
    }

    int calculateYearsTillRetirement(){
        return 60-age;
    }

    int getWage(){
        return wage;
    }

    int getAge(){
        return age;
    }

    int getExperience(){
        return experience;
    }
}

```

Slika 2 Primer definisanja interfejsa i klase koja implementira taj interfejs

Klasa "Person" implementira interfejs "Employee" i u obavezi je da implementira njene metode. Ukoliko ne implementira neku od metoda dolazi do semantičke greške. Na slici takodje vidimo primer deklaracije jednostavnih get metoda za attribute ove klase. Na Slici 3 je prikazan način upotrebe ove klase i pozivi njenih metoda.

```
int main() {  
    int age;  
    int wage;  
    int experience;  
    int vacationDays;  
    Person p = new Person(1000, 22, 1);  
    age = p.getAge();  
    wage = p.getWage();  
    experience = p.getExperience();  
    vacationDays = p.calculateVacationDays();  
}
```

Slika 3 Primer instanciranja klase koja implementira interfejs i pozivi njenih metoda

Na Slici 4 prikazan je primer upotrebe atributa i gettera za postavljanje vrednosti atributa izmedju objekata.

```
int main() {  
    Person p = new Person();  
    Person p1 = new Person(32, 2000, 11);  
    p.wage = p1.wage;  
    p.age = p1.getAge();  
}
```

Slika 4 Primer interakcije izmedju objekata

4. Ideje za nastavak

U projektu nije podržan pristup atributima ugnježđenih objekata (Objekat.objekat.atribut...) i prosledjivanje objekata u konstruktoru. Takodje, kao ideju za budući rad izdvojio bih generisanje koda koje bi predstavljalo veliki izazov za ovu temu.

5. Literatura

1. <https://www.geeksforgeeks.org/classes-objects-java/>
2. <https://www.geeksforgeeks.org/interfaces-in-java/>