

## Kreiranje projekta u Eclipse okruženju

Potrebno je napraviti projekat sa sledećim nazivom: **Prezime\_Ime\_broj-indeksa\_grupaZadataka**, gde je broj indeksa u formatu četiri cifre za godinu – četiri cifre za broj indeksa (Na primer: Stojimirović\_Dejan\_2014-0016\_2).

Nakon toga, u paketu (**eng. package**) **src** je potrebno da kreiramo paket **main**. U tom paketu se kreiraju sve klase koje su navedene u tekstu zadatka, kao na primer klase JSLista, DSLista, Niz,...). Veoma je bitno da se klasa zove onako kako je navedeno u tekstu zadatka.

**VEOMA JE BITNO URADITI KAKO JE NAVEDENO U UPUTSTVU KAKO BI RAD MOGAO BITI PREGLEDAN!**

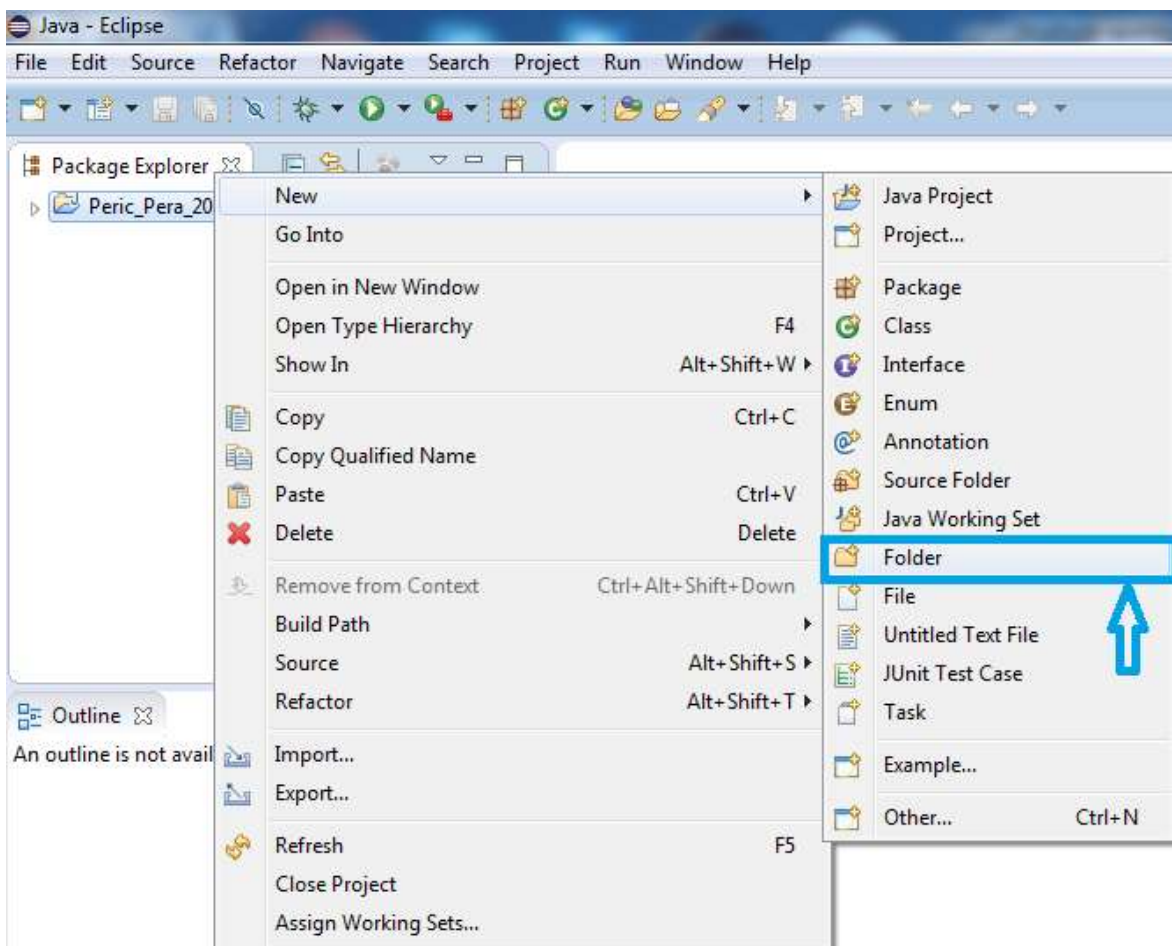
## Dodavanje biblioteke u Java projekat u Eclipse razvojnom okruženju

Nakon što je kreiran projekat u Eclipse-u, potrebno je napraviti folder u okviru projekta. To se može uraditi na dva načina:

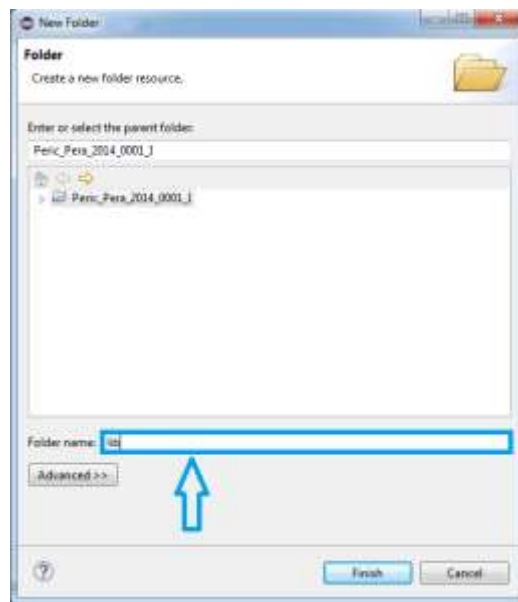
### Dodavanje novog foldera

#### NAČIN 1: Pomoću Eclipse razvojnog okruženja

1. Desni klik na projekat -> **New** -> **Folder**



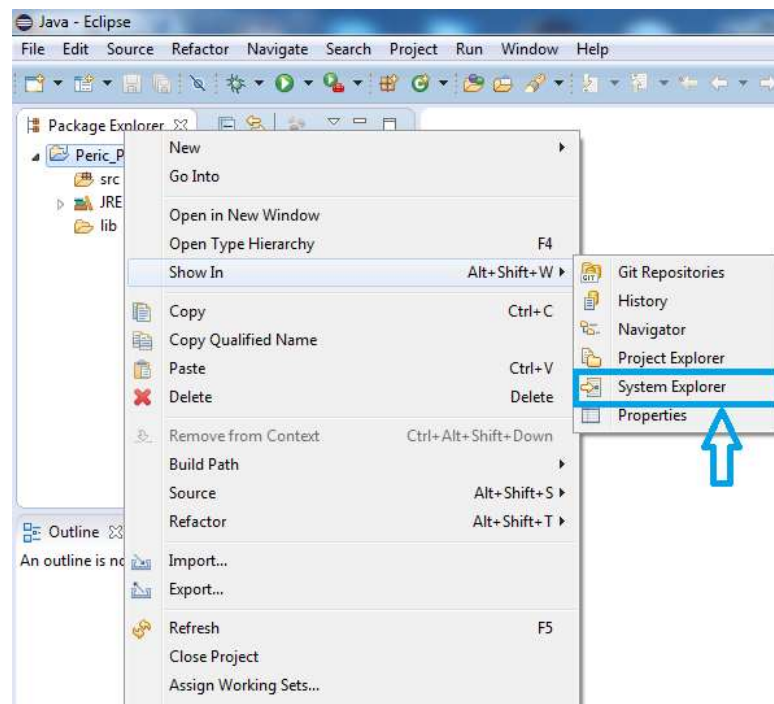
2. Nakon toga, dobija se prozor kao na sledećoj slici. Potrebno je uneti ime foldera **lib**.



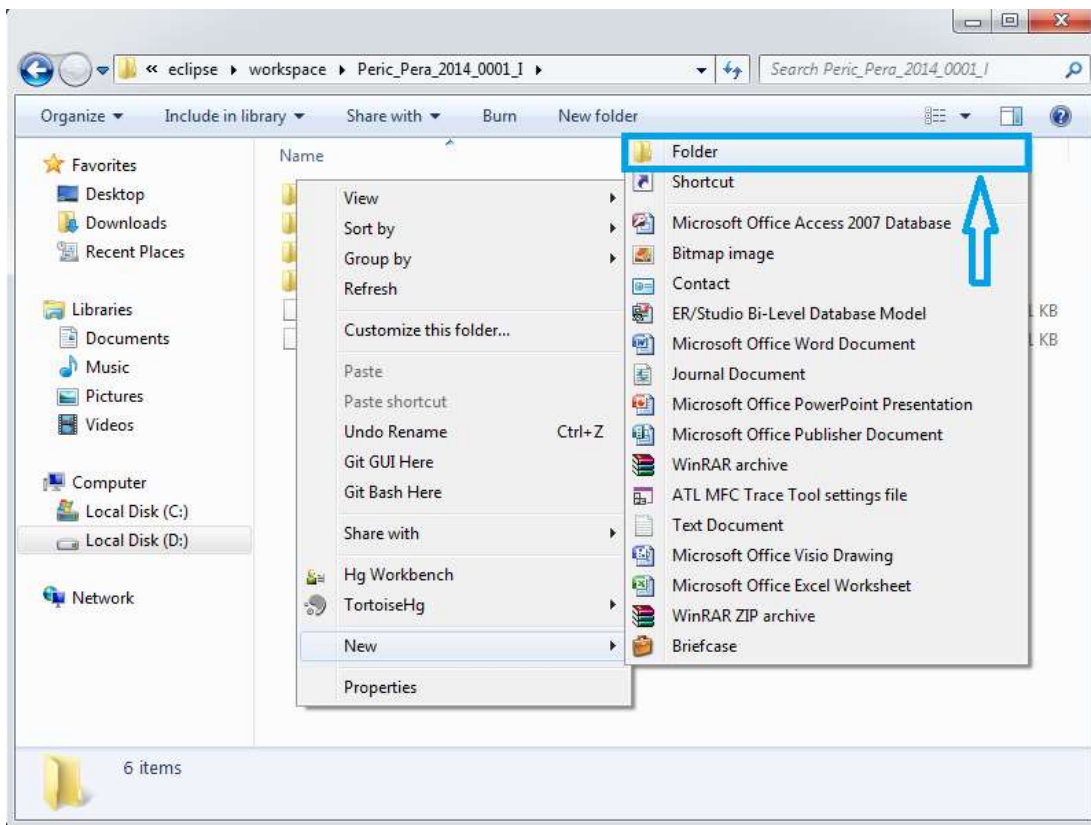
3. Potrebno je kliknuti na dugme „**Finish**“

## NAČIN 2: Pomoću File sistema

1. Kliknuti desni klik na projekat -> **Show In** -> **System Explorer**

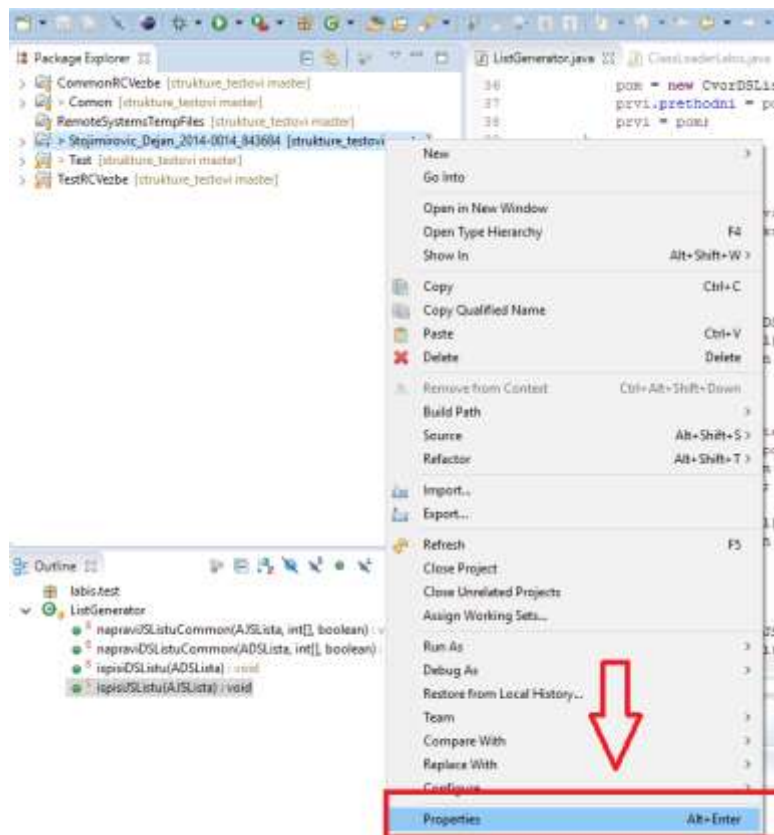


2. Nakon toga, otvoriće se folder sa projektima. Potrebno je naći svoj projekat i duplim klikom na njega otvoriti folder, te u njemu dodati folder desnim klikom u prazno i klikom na **New >Folder**

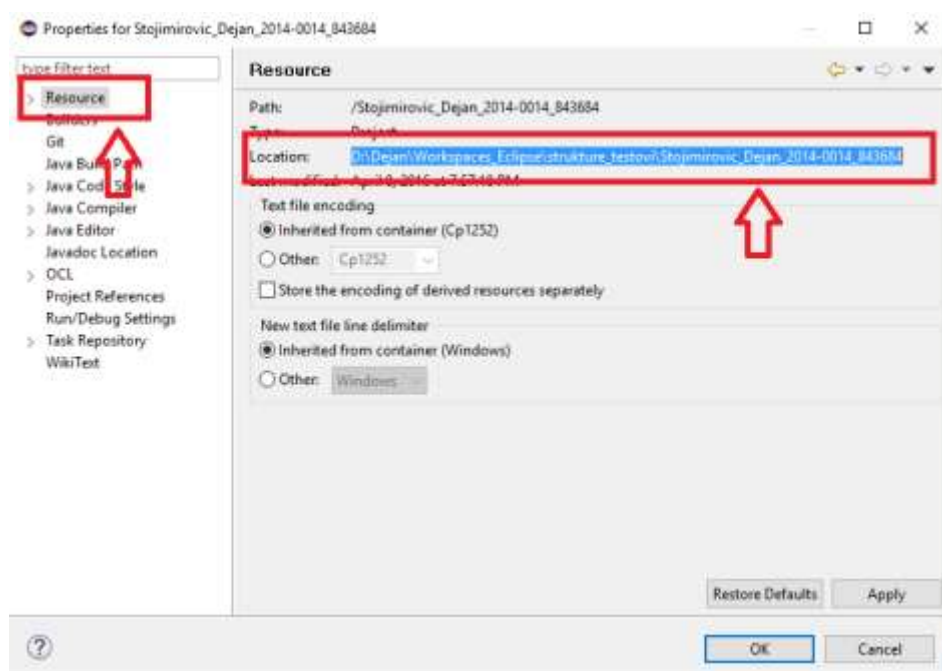


3. Potrebno je imenovati folder **lib**.

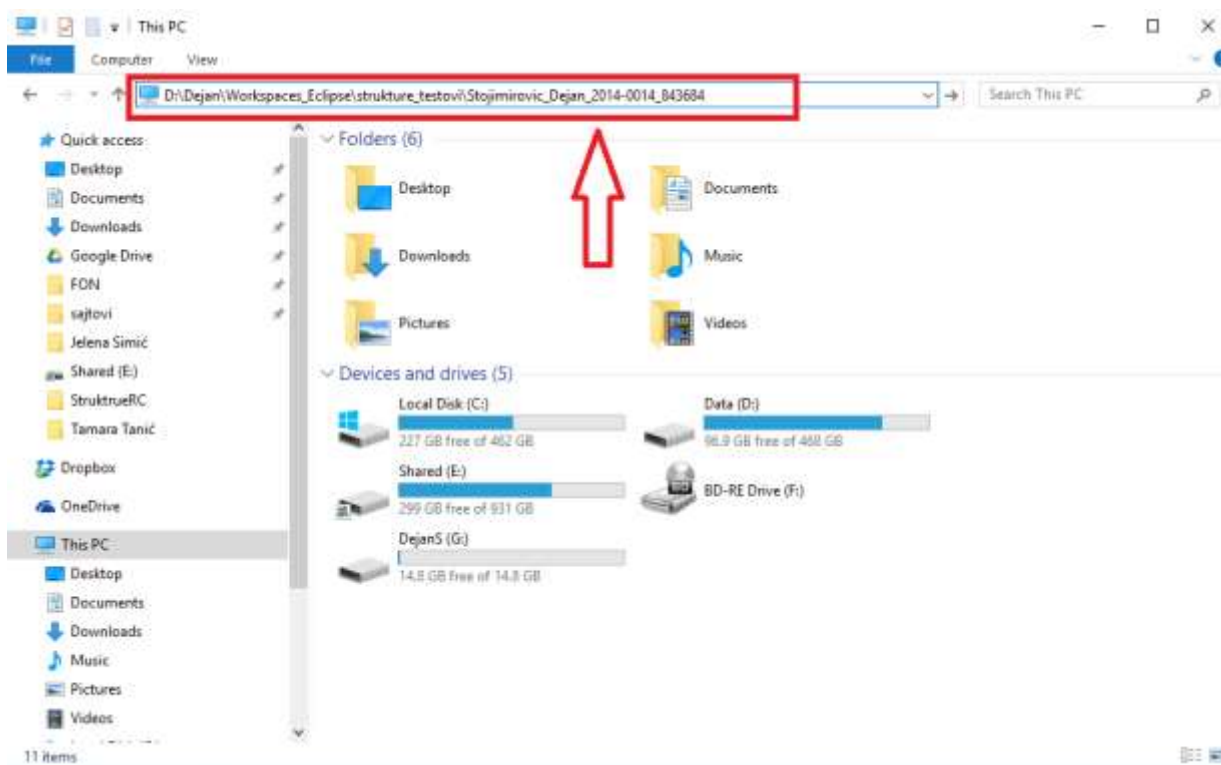
U slučaju da nemate opciju **System Explorer**, kliknete desnim klikom na **lib folder** i odaberete opciju **Properties**.



Kliknete na opciju **resource** i kopirate adresu do foldera.



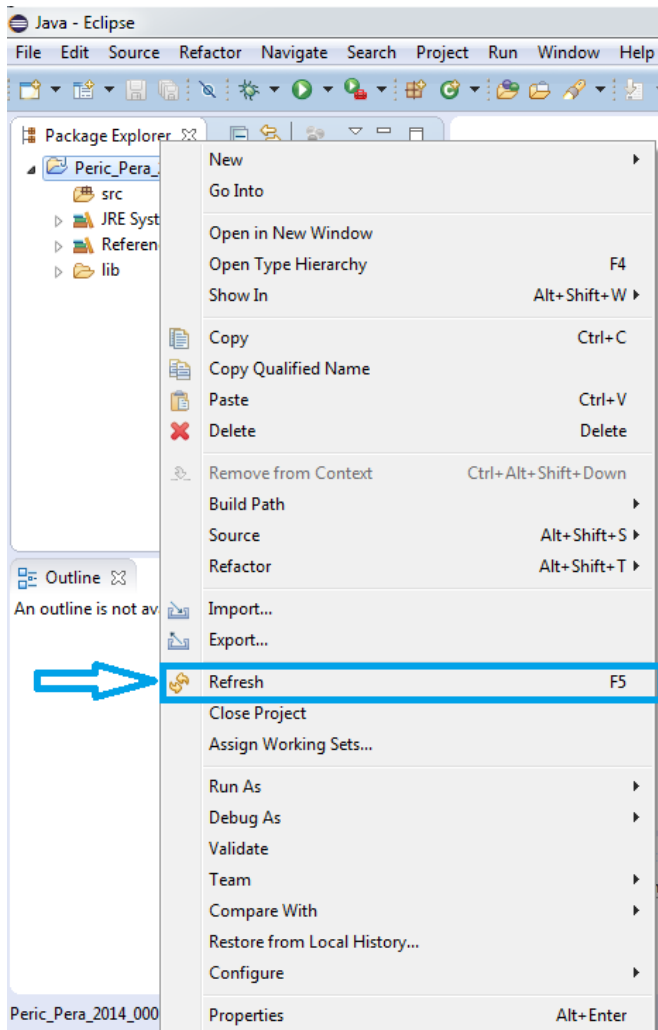
Nakon toga otvorimo **My Computer**, u prostoru za adresu kopiramo putanju do lib foldera i pritisamo enter.



## Dodavanje biblioteke

Nakon pravljenja foldera, potrebno je kopirati biblioteku koja je data na raspolaganje u taj folder (**lib**).

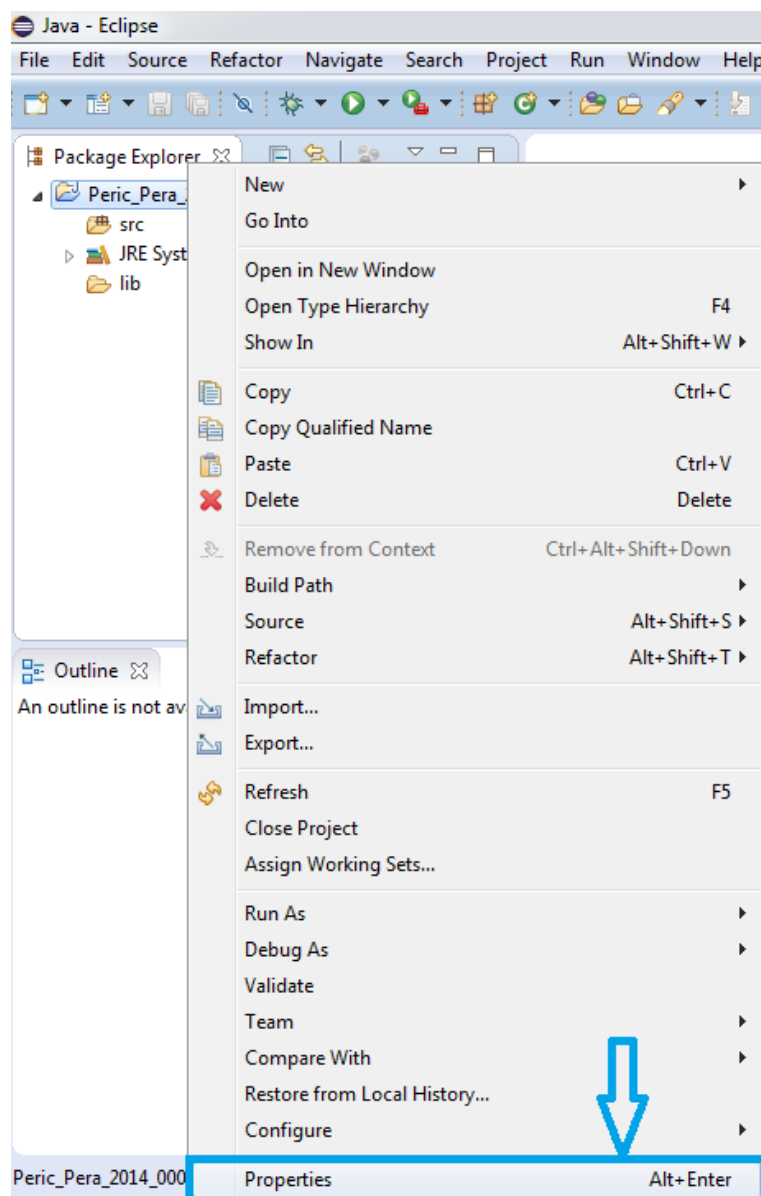
U Eclipse-u, potrebno je kliknuti desni klik na projekat i odabrati opciju **Refresh**.



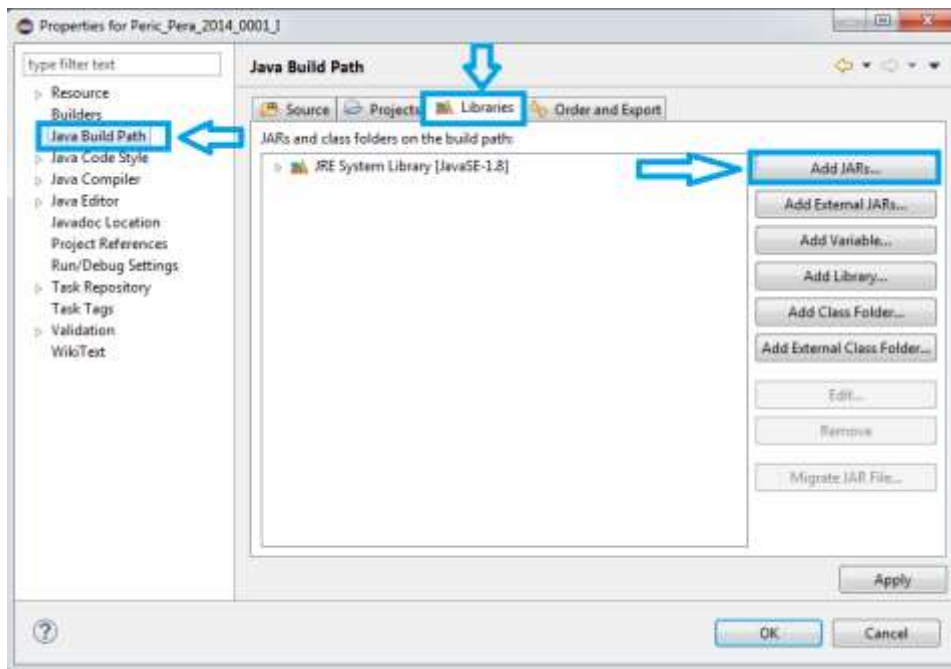
## Učitavanje biblioteke

Potrebno je postaviti referencu na biblioteku. To se može uraditi na sledeći način:

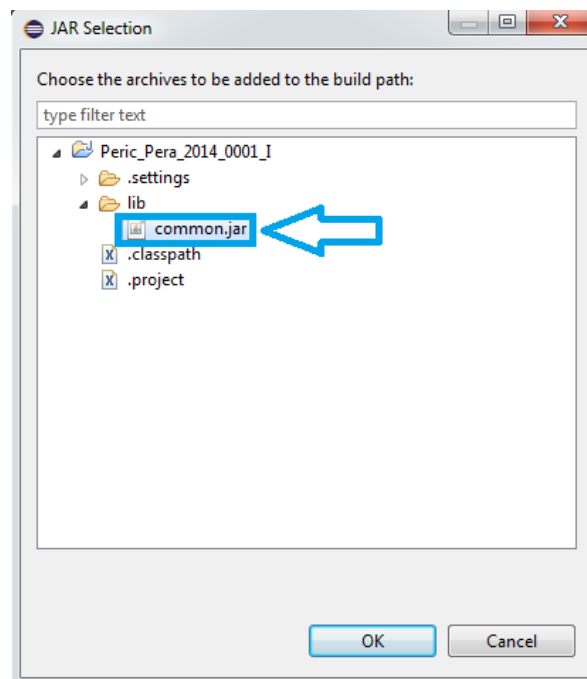
Desni klik na projekat -> **Properties**



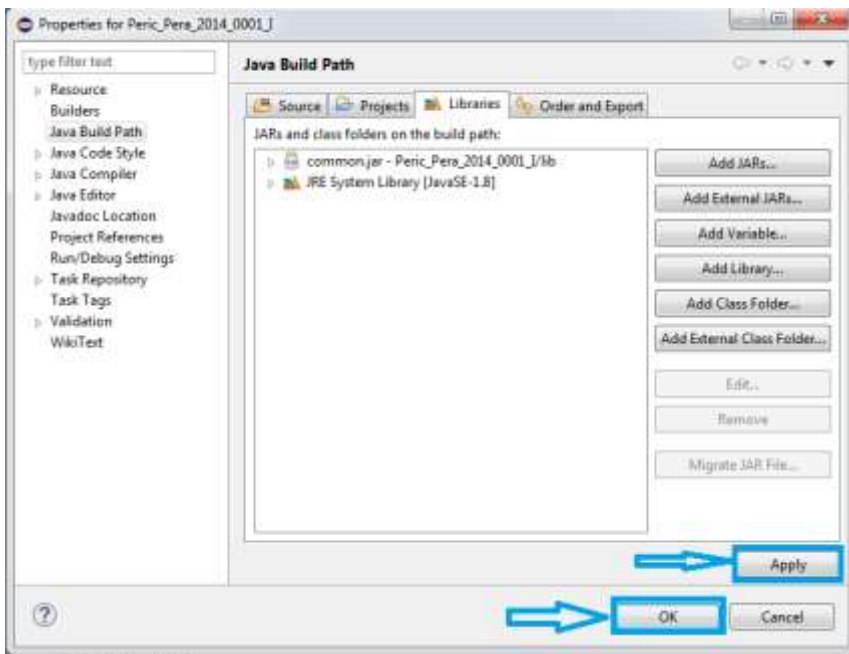
Sa leve strane se odabere opcija **Java Build Path**. U centralnom delu prozora pojaviće se četiri kartice. Potrebno je odabrati karticu **Libraries**, i nakon toga se desne strane kliknuti na dugme „**Add JARs...**“. To je prikazano na slici 6.



Klikom na dugme „**Add JARs...**“ Otvara se prozor prikazan na slici 7. Potrebno je označiti biblioteku koja se nalazi u novokreiranom folderu **lib**.



Klikom na dugme „**OK**“ potvrđuje se odabir. Nakon ovoga, potrebno je kliknuti na dugme „**Apply**“ i potom na dugme „**OK**“, i time je postavljena referenca na biblioteku.





## Uputstvo za korišćenje klase ListGenerator

Radi potreba testiranja algoritama, pomoću metoda klase ListGenerator, moguće je napraviti listu. Klasa ListGenerator ima dve metode koje su statičke (pozivaju se sa nivoa klase) :

### ***ListGenerator.napraviJSListuCommon(lista, elementi, ciklicna)***

Ova metoda kao parametar prima objekat klase JSListu, prima niz integer-a koji će biti vrednosti čvorova liste, i kao treći parametar prima boolean vrednost koji se odnosi na to da li je lista ciklična (**true** ako jeste, **false** ako nije). Dat je primer na slici ispod.

```
*Test.java JSListu.java
1 import labis.test.ListGenerator;
2
3 public class Test {
4     public static void main(String[] args) {
5
6         JSListu lista = new JSListu();
7         int[] niz = new int[]{4,5,7,8,-4,11};
8         ListGenerator.napraviJSListuCommon(lista, niz, false);
9     }
10 }
```

Druga metoda je:

### ***ListGenerator.napraviDSLListuCommon(lista, elementi, ciklicna)***

Ova metoda kao parametar prima objekat klase DSLListu, prima niz integer-a koji će biti vrednosti čvorova liste, i kao treći parametar prima boolean vrednost koji se odnosi na to da li je lista ciklična (true ako jeste, false ako nije). Dat je primer na slici ispod.

```
Test.java DSLListu.java
1 import labis.test.ListGenerator;
2
3 public class Test {
4     public static void main(String[] args) {
5
6         DSLListu lista = new DSLListu();
7         int[] niz = new int[] { 4, 5, 7, 8, -4, 11 };
8         ListGenerator.napraviDSLListuCommon(lista, niz, false);
9     }
10 }
11 }
```

**NAPOMENA:** Ukoliko je potrebno napraviti praznu listu, nije potrebno pozivati prikazane metode, dovoljno je napraviti objekat klase JSListu ili DSLListu.

Na ovaj način, dobija se lista sa onoliko čvorova koliko elemenata ima dati niz, i nad objektom klase JSListu ili DSLListu (u primerima objekat lista) mogu se pozivati metode koje su implementirane i testirati da li se lista ponaša očekivano. Primer je dat na sledećoj slici.

```
Test.java x DSLista.java
1 import labis.exception.LabisException;
2 import labis.test.ListGenerator;
3
4 public class Test {
5     public static void main(String[] args) {
6
7         DSLista lista = new DSLista();
8         int[] niz = new int[] { 4, 5, 7, 8, -4, 11 };
9         ListGenerator.napraviDSLlistuCommon(lista, niz, false);
10
11         try {
12             lista.popuniListu();
13         } catch (LabisException e) {
14             // TODO Auto-generated catch block
15             e.printStackTrace();
16         }
17     }
18 }
```

Takođe, moguće je koristiti i metode:

1. *ListGenerator.ispisiDSLlistu(prvi);*
2. *ListGenerator.ispisiJSListu(prvi);*

Prikazane metode ispisuju na konzoli listu koja im je prosleđena kao parametar. Primer je dat na slici ispod.

```
public static void main(String[] args) {
    JSList lista = new JSList();
    ListGenerator.napraviJSListuCommon(lista, new int[] { 4, 2, 6 }, false);

    try {
        lista.izbaciSaKraja();
        ListGenerator.ispisiJSListu(lista.prvi);
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```