

1. Classes e Objetos

1.1 Classe Pessoa

```
● ● ●

1  public abstract class Pessoa {
2      String nome;
3      String cpf;
4      int idade;
5
6      Pessoa(String nome, String cpf, int idade){
7          this.nome = nome;
8          this.cpf = cpf;
9          this.idade = idade;
10     }
11
12     public String getNome(){
13         return nome;
14     }
15
16     public void setNome(String nome){
17         this.nome = nome;
18     }
19
20     public String getCpf(){
21         return cpf;
22     }
23
24     public void setCpf(String cpf){
25         this.cpf = cpf;
26     }
27
28     public int getIdade(){
29         return idade;
30     }
31
32     public void setIdade(int idade){
33         this.idade = idade;
34     }
35
36     public void apresentar(){
37         System.out.println(
38             "Nome: " + nome +
39             " | CPF: " + cpf +
40             " | Idade: " + idade
41         );
42     }
43 }
```

Na classe Pessoa, ela foi definida como uma classe abstrata, ela é um molde para as demais classes.

```
● ● ●  
1  public abstract class Pessoa {  
2      String nome;  
3      String cpf;  
4      int idade;  
5  
6      Pessoa(String nome, String cpf, int idade){  
7          this.nome = nome;  
8          this.cpf = cpf;  
9          this.idade = idade;  
10     }  
11 }
```

Como ela é uma classe abstrata ela não pode ser instanciada então criamos um método construtor, para definir os objetos.

```
11  
12     public String getNome(){  
13         return nome;  
14     }  
15  
16     public void setNome(String nome){  
17         this.nome = nome;  
18     }  
19  
20     public String getCpf(){  
21         return cpf;  
22     }  
23  
24     public void setCpf(String cpf){  
25         this.cpf = cpf;  
26     }  
27  
28     public int getIdade(){  
29         return idade;  
30     }  
31  
32     public void setIdade(int idade){  
33         this.idade = idade;  
34     }  
35 }
```

Métodos de acesso dos objetos

Get e Set

Foi criado um método para mostrar no terminal as informações fornecidas nos métodos de acesso.

```
35  
36     public void apresentar(){  
37         System.out.println(  
38             "Nome: " + nome +  
39             " | CPF: " + cpf +  
40             " | Idade: " + idade  
41         );  
42     }  
43 }
```

1.2 Classe Disciplina

Na classe Disciplina, assim como na classe Pessoa nós temos um método construtor mas a adição dos dados no caso criando uma disciplina.

```
● ● ●
1 public class Disciplina {
2     String codigo;
3     String nome;
4     int semestre;
5
6     Disciplina(String codigo, String nome, int semestre){
7         this.codigo = codigo;
8         this.nome = nome;
9         this.semestre = semestre;
10    }
11
12    @Override
13    public String toString(){
14        return "Disciplina: " + nome + " (Codigo: " + codigo + ")";
15    }
16 }
```

Mas temos um código a mais no método `toString`, porque adicionamos isso? para que quando a classe Disciplina fosse chamada ela mostre o que foi definido, e não, onde ela está alocado no computador

```
11
12    @Override
13    public String toString(){
14        return "Disciplina: " + nome + " (Codigo: " + codigo + ")";
15    }
16 }
```