

Fashion_Shows-MFES

December 30, 2017

Contents

1 Designer	1
2 Event	2
3 FashionFestival	5
4 FashionUser	7
5 Model	10
6 MyTests	11
7 Runway	12
8 TestApp	13
9 Tests	16

1 Designer

```
class Designer

types
  public String = seq of char;

instance variables
  public name : String;
  public age : String;
  public nationality : String;
  public address : String;
  public style : String;
  public output : String := "";

inv len name > 0 and len age > 0 and len nationality > 0 and len address > 0 and len style > 0;

operations

  public Designer :
    String *
    String *
    String *
    String *
```

```

        String ==> Designer
Designer(nm, ag, nt , ad, sty) ==
(
    name := nm;
    age := ag;
    nationality := nt;
    address := ad;
    style := sty;
    return self;
);

--retorna os parametros da class event

public pure getName : () ==> String
getName() == return name
post RESULT = name;

public pure getAge : () ==> String
getAge() == return age
post RESULT = age;

public pure getNationality : () ==> String
getNationality() == return nationality
post RESULT = nationality;

public pure getAddress : () ==> String
getAddress() == return address
post RESULT = address;

public pure getStyle : () ==> String
getStyle() == return style
post RESULT = style;

public printDesigner: () ==> String
printDesigner() == (
output := "Designer Name: "^name^"\n"
        ^"Age: "^age^"\n"
        ^"Nationality: "^nationality^"\n"
        ^"Address: "^address^"\n"
        ^"Style: "^style^"\n";
return output;
)
post RESULT = output;

end Designer

```

Function or operation	Line	Coverage	Calls
Designer	17	100.0%	2
getAddress	46	100.0%	2
getAge	38	100.0%	2
getName	34	100.0%	6
getNationality	42	100.0%	2

getStyle	50	100.0%	2
printDesigner	54	100.0%	1
Designer.vdmpp		100.0%	17

2 Event

```

class Event

types
  public String = seq of char;

instance variables
  public name : String := "";
  public date : String := "";
  public local : String := "";
  public time : String := "";
  public duration : String := "";
  public theme : String := "";
  public gender: String := "";
  public collection: String := "";
  public output: String := "";
  --Lista de desfiles
  private runways: seq of Runway := [];

  inv len name > 0 and len date > 0 and len local > 0 and len time > 0 and len duration > 0 and
    len theme > 0 and len gender > 0 and len collection > 0;

operations

  public Event :
    String *
    String *
    String *
    String *
    String *
    String *
    String *
    String ==> Event
  Event(nm, dt, lc, hr , dr, tm, gr, cl) == (
    name := nm;
    date := dt;
    local := lc;
    time := hr;
    duration := dr;
    theme := tm;
    gender := gr;
    collection := cl;
    return self
  );

  --retorna os parametros da class event

  public pure getName : () ==> String
    getName() == return name
    post RESULT = name;

  public pure getDate : () ==> String
    getDate() == return date

```

```

post RESULT = date;

public pure getLocal : () ==> String
  getLocal() == return local
post RESULT = local;

public pure getTime : () ==> String
  getTime() == return time
post RESULT = time;

public pure getDuration : () ==> String
  getDuration() == return duration
post RESULT = duration;

public pure getTheme : () ==> String
  getTheme() == return theme
post RESULT = theme;

public pure getGender : () ==> String
  getGender() == return gender
post RESULT = gender;

public pure getCollection : () ==> String
  getCollection() == return collection
post RESULT = collection;

public pure getRunways : () ==> seq of Runway
  getRunways() == return runways
post RESULT = runways;

public pure getNumberRunways : () ==> nat
  getNumberRunways() == return len runways
post RESULT = len runways;

public insertRunway : Runway ==> ()
  insertRunway(r) ==
  (
    runways := runways ^ [r];
  )
  pre r not in set elems getRunways()
  post r in set elems getRunways();

public printEvent : () ==> String
  printEvent() == (
    output := "Event Name: "^name^"\n"
             ^"Date: "^date^"\n"
             ^"Time: "^time^"\n"
             ^"Theme: "^theme^"\n"
             ^"Gender: "^gender^"\n"
             ^"Collection: "^collection^"\n";
    return output;
  )
post RESULT = output;

```

end Event

Function or operation	Line	Coverage	Calls
Event	24	100.0%	3
getCollection	74	100.0%	2
getDate	50	100.0%	2
getDuration	62	100.0%	1
getGender	70	100.0%	2
getLocal	54	100.0%	1
getName	46	100.0%	15
getNumberRunways	82	100.0%	3
getRunways	78	100.0%	15
getTheme	66	100.0%	2
getTime	58	100.0%	2
insertRunway	86	100.0%	4
printEvent	95	100.0%	1
Event.vdmpp		100.0%	53

3 FashionFestival

```
class FashionFestival

types
  public String = seq of char;

instance variables
  public name : String := "";
  public dateBegin : String := "";
  public dateEnd : String := "";
  public local : String := "";
  public events: seq of Event := [];
  public numberEvents: int := 0;
  public fashionUsers : set of FashionUser := {};
  public output : String := "";
  public outputN : String := "";

  inv len name > 0 and len dateBegin > 0 and len dateEnd > 0 and len local > 0;

operations

  public FashionFestival :
    String *
    String *
    String *
    String
    ==> FashionFestival
  FashionFestival(nm, di, df , lc) ==
  (
    name := nm;
    dateBegin := di;
    dateEnd := df;
    local := lc;
    return self;
```

```

);

--retorna os parametros da class event

public pure getName : () ==> String
  getName() == return name
post RESULT = name;

public pure getDateBegin : () ==> String
  getDateBegin() == return dateBegin
post RESULT = dateBegin;

public pure getDateEnd : () ==> String
  getDateEnd() == return dateEnd
post RESULT = dateEnd;

public pure getLocal : () ==> String
  getLocal() == return local
post RESULT = local;

public pure getEvents : () ==> seq of Event
  getEvents() == return events
post RESULT = events;

--Adiciona designer ao evento

public insertEvent : Event ==> ()
  insertEvent(ev) ==
  (
    events := [ev] ^ events;
    numberEvents := numberEvents + 1;
  )
pre ev not in set elems getEvents()
post ev in set elems getEvents();

--Adiciona designer ao evento

public insertFashionUser: (FashionUser) ==> ()
  insertFashionUser(us) ==
  (
    fashionUsers := fashionUsers union {us};
  )
pre us not in set getFashionUsers()
post us in set getFashionUsers();

--retorna nr designers do evento

public pure getNumberEvents : () ==> int
  getNumberEvents() == return numberEvents
post RESULT = numberEvents;

--retorna utilizadores da aplicacao

public pure getFashionUsers : () ==> set of FashionUser
  getFashionUsers() == return fashionUsers
post RESULT = fashionUsers;

public pure getNumberFashionUsers: () ==> nat
  getNumberFashionUsers() ==
  return (card fashionUsers)

```

```

post RESULT = (card fashionUsers);

public printFashionFestival: () ==> String
printFashionFestival() == (
output := "Name: "^name^"\n"
        ^"Date Begin: "^dateBegin^"\n"
        ^"Date End: "^dateEnd^"\n"
        ^"Local: "^local^"\n";
return output;
)
post RESULT = output;

public printFashionFestivalName: () ==> String
printFashionFestivalName() == (
outputN := name^"\n";
return outputN;
)
post RESULT = outputN;
end FashionFestival

```

Function or operation	Line	Coverage	Calls
FashionFestival	20	100.0%	2
getDateBegin	40	100.0%	2
getDateEnd	44	100.0%	2
getEvents	52	100.0%	21
getFashionUsers	81	100.0%	6
getLocal	48	100.0%	2
getName	36	100.0%	5
getNumberEvents	76	100.0%	2
getNumberFashionUsers	85	100.0%	1
insertEvent	57	100.0%	4
insertFashionUser	67	100.0%	2
printFashionFestival	91	100.0%	1
printFashionFestivalName	102	100.0%	2
FashionFestival.vdmpp		100.0%	52

4 FashionUser

```

class FashionUser

types
public String = seq of char;

instance variables
private username: String;
private password: String;
public name: String;
public age: String;

```

```

--designers favoritos
private designers: seq of Designer := [];
private numberDesigners: int := 0;
--modelos favoritos
private models: seq of Model := [];
private numberModels: int := 0;
--eventos que utilizador vai
private events: seq of Event := [];
private numberEvents: int := 0;
private output : String := "";

inv len username > 0 and len password = 6 and len name > 0 and len age > 0;

operations

public FashionUser : String *
    String *
    String *
    String ==> FashionUser
FashionUser(u, p, nm, ag) ==
(
    username := u;
    password := p;
    name := nm;
    age := ag;
    return self;
);

--retorna os parametros da class fashionUser

public pure getUsername : () ==> String
getUsername() == return username
post RESULT = username;

public pure getPassword : () ==> String
getPassword() == return password
post RESULT = password;

public pure getName : () ==> String
getName() == return name
post RESULT = name;

public pure getAge : () ==> String
getAge() == return age
post RESULT = age;

--DESIGNERES FAVORITOS
--Adiciona designer favorito ao utilizador

public insertDesigner : Designer ==> ()
insertDesigner(d) ==
(
    numberDesigners := numberDesigners + 1;
    designers := designers ^ [d];
)
pre d not in set elems designers
post d in set elems designers;

--retorna nr designers favoritos

```



```

public pure getNumberFavDesigners : () ==> nat
getNumberFavDesigners() == return numberDesigners
post RESULT = numberDesigners;

public getDesigners: () ==> seq of Designer
getDesigners() == return designers
post RESULT = designers;

public insertModel : Model ==> ()
insertModel(d) ==
(
  numberModels := numberModels + 1;
  models := models ^ [d];
)
pre d not in set elems models
post d in set elems models;

--retorna nr designers favoritos

public pure getNumberFavModels : () ==> nat
getNumberFavModels() == return numberModels
post RESULT = numberModels;

public getModels: () ==> seq of Model
getModels() == return models
post RESULT = models;

--EVENTOS DO UTILIZADOR
--Adiciona evento

public insertEvent : Event ==> ()
insertEvent(ev) ==
(
  numberEvents := numberEvents + 1;
  events := events ^ [ev];
)
pre ev not in set elems events
post ev in set elems events;

--Retorna nr designers favoritos

public pure getNumberEvents : () ==> nat
getNumberEvents() == return numberEvents
post RESULT = numberEvents;

public getEvents: () ==> seq of Event
getEvents() == return events
post RESULT = events;

public printUser: () ==> String
printUser() == (
output := "Username: "^username^"\n"
        ^"Password: "^password^"\n"
        ^"Name: "^name^"\n"
        ^"Age: "^age^"\n";
return output;
)
post RESULT = output;

```

```
end FashionUser
```

Function or operation	Line	Coverage	Calls
FashionUser	26	100.0%	3
getAge	52	100.0%	2
getDesigners	73	100.0%	2
getEvents	113	100.0%	2
getModels	91	100.0%	2
getName	48	100.0%	2
getNumberEvents	108	100.0%	1
getNumberFavDesigners	69	100.0%	1
getNumberFavModels	87	100.0%	1
getPassword	44	100.0%	2
getUsername	40	100.0%	2
insertDesigner	59	100.0%	3
insertEvent	97	100.0%	4
insertModel	77	100.0%	6
printUser	117	100.0%	1
FashionUser.vdmpp		100.0%	34

5 Model

```
class Model

types
  public String = seq of char;

instance variables
  public name : String;
  public age : String;
  public nationality : String;
  public address : String;
  public output : String := "";

  inv len name > 0 and len age > 0 and len nationality > 0 and len address > 0;

operations

  public Model :
    String *
    String *
    String *
    String ==> Model
  Model(nm, ag, nt , ad) ==
  (
    name := nm;
    age := ag;
    nationality := nt;
    address := ad;
    return self;
  );
```

```

--retorna os parametros da class Model

public pure getName : () ==> String
  getName() == return name
  post RESULT = name;

public pure getAge : () ==> String
  getAge() == return age
  post RESULT = age;

public pure getNationality : () ==> String
  getNationality() == return nationality
  post RESULT = nationality;

public pure getAddress : () ==> String
  getAddress() == return address
  post RESULT = address;

public printModel: () ==> String
  printModel() == (
    output := "Model Name: "^name^"\n"
             ^"Age: "^age^"\n"
             ^"Nationality: "^nationality^"\n"
             ^"Address: "^address^"\n";
  )
  return output;
  post RESULT = output;
end Model

```

Function or operation	Line	Coverage	Calls
Model	16	100.0%	4
getAddress	43	100.0%	2
getAge	35	100.0%	2
getName	31	100.0%	10
getNationality	39	100.0%	2
printModel	47	100.0%	1
Model.vdmpp		100.0%	21

6 MyTests

```

class MyTests

operations

  -- Simulates assertion checking by reducing it to pre-condition checking.
  -- If 'arg' is null or undefined, a pre-condition violation will be signaled.
  -- Verification of pre-conditions must be enabled in order for this to work

  protected assertNotNull : ? ==> ()

```

```

assertNotNull(arg) == return
pre arg <> nil and arg <> undefined;

-- Simulates assertion checking by reducing it to post-condition checking.
-- If values are not equal, prints a message and generates a post-conditions violation.

protected assertEquals : ? * ? ==> ()
assertEquals(expected, actual) ==
  if expected <> actual then
    (
      IO'print("Actual value (");
      IO'print(actual);
      IO'print(") different from expected (");
      IO'print(expected);
      IO'println(")\n")
    )
  post expected = actual;
end MyTests

```

Function or operation	Line	Coverage	Calls
assertEquals	14	38.8%	0
assertNotNull	8	100.0%	13
MyTests.vdmpp		57.6%	13

7 Runway

```

class Runway

types
  public String = seq of char;

instance variables
  public name : String := "";
  private designers: set of Designer := {};
  private numberDesigners: int := card designers;
  private models: seq of Model := [];
  private numberModels: int := 0;
  private output : String := "";

  inv len name > 0;

operations

  public Runway :
    (String)
    ==> Runway
  Runway(nm) == (
    name := nm;
    return self
  );

  public pure getName : () ==> String
    getName() == return name
  post RESULT = name;

```

```

public pure getModels : () ==> seq of Model
  getModels() == return models
post RESULT = models;

public pure getDesigners : () ==> set of Designer
  getDesigners() == return designers
post RESULT = designers;

public pure getDesignersNumber : () ==> nat
  getDesignersNumber() == return card designers
post RESULT = (card designers);

public insertDesigner : Designer ==> ()
  insertDesigner(dg) ==
  (
    designers := designers union {dg};
  )
  pre dg not in set getDesigners()
  post dg in set getDesigners();

  --MODELOS DO DESFILE
  --Adiciona model ao desfile

public insertModel : Model ==> ()
  insertModel(md) ==
  (
    numberModels := numberModels + 1;
    models := models ^ [md];
  )
  pre md not in set elems getModels()
  post md in set elems getModels();

  --retorna nr modelso do desfile

public pure getNumberModels : () ==> int
  getNumberModels() == return numberModels
post RESULT = numberModels;

public printRunway: () ==> String
  printRunway() == (
    output := name^"\n";
    return output;
  )
post RESULT = output;

end Runway

```

Function or operation	Line	Coverage	Calls
Runway	17	100.0%	3
getDesigners	33	100.0%	10
getDesignersNumber	37	100.0%	1
getModels	29	100.0%	12
getName	25	100.0%	6

getNumberModels	61	100.0%	1
insertDesigner	41	100.0%	8
insertModel	51	100.0%	5
printRunway	65	100.0%	1
Runway.vdmpp		100.0%	47

8 TestApp

```

class TestApp

types
public String = seq of char;

instance variables

operations

public static printTests: () ==> ()
printTests() ==
(
    new Tests().run();
);

public static getUsers: () ==> set of FashionUser
getUsers() ==
(
    return Tests`getAppUsers();
);

public static getFestivals: () ==> seq of FashionFestival
getFestivals() ==
(
    return Tests`getFestivals();
);

--Nomes dos festivais disponiveis

public static getFestivalsNames: () ==> String
getFestivalsNames() ==
(
    for counter = 1 to len Tests`getFestivals() do (
        IO`print(counter);
        return (Tests`getFestivals() (counter)).printFashionFestivalName();
    );
);

--Festival seleccionado apos isto imprimirs

public static getFestival: (int) ==> FashionFestival
getFestival(optionFestival) ==
(
    return (Tests`getFestivals() (optionFestival));
);

```

```

--eventos do festival selecionado

public static getFestivalEvents: (int) ==> seq of Event
getFestivalEvents(optionFestival) ==
(
return (Tests`getFestivals() (optionFestival)).getEvents();
);

public static getFestivalUsers: (int) ==> set of FashionUser
getFestivalUsers(optionFestival) ==
(
return (Tests`getFestivals() (optionFestival)).getFashionUsers();
);

--nomes dos eventos do festival selecionado

public static getFestivalEventsNames: (int) ==> String
getFestivalEventsNames(optionFestival) ==
(
for counter = 1 to len (Tests`getFestivals() (optionFestival)).getEvents() do (
IO`print(counter);
return (((Tests`getFestivals() (optionFestival)).getEvents() (counter)).getName());
);
);

--Evento selecionado do festival selecionado

public static getEvent: (int) * (int) ==> Event
getEvent(optionFestival,optionEvent) ==
(
return (getFestivalEvents(optionFestival) (optionEvent));
);

--Retorna desfiles do evento selecionado do festival selecionado

public static getRunwaysByEvent : (int) * (int) ==> seq of Runway
getRunwaysByEvent(optionFestival, optionEvent) ==
(
return ((getFestivalEvents(optionFestival) (optionEvent)).getRunways());
);

public static getRunwaysNames: (int) * (int) ==> String
getRunwaysNames(optionFestival,optionEvent) ==
(
for counter = 1 to len getRunwaysByEvent(optionFestival,optionEvent) do (
IO`print(counter);
return ((getRunwaysByEvent(optionFestival,optionEvent) (counter)).getName());
);
);

--Retorna desfile selecionado do evento selecionado do festival selecionado

public static getOneRunwayByEvent : (int) * (int) * (int) ==> Runway
getOneRunwayByEvent(optionFestival, optionEvent, optionRunWay) ==
(
return (getRunwaysByEvent(optionFestival,optionEvent)) (optionRunWay);
);

--Retorna modelos do desfile selecionado do evento selecionado do festival selecionado

public static getModelsByRunway : (int) * (int) * (int) ==> seq of Model

```

```

getModelsByRunway(optionFestival, optionEvent, optionRunWay) ==
(
return getOneRunwayByEvent(optionFestival,optionEvent,optionRunWay).getModels();
);

--Retorna designers do desfile selecionando do evento selecionado do festival selecionado

public static getDesignersByRunway : (int) * (int) * (int) ==> set of Designer
getDesignersByRunway(optionFestival, optionEvent, optionRunWay) ==
(
return getOneRunwayByEvent(optionFestival,optionEvent,optionRunWay).getDesigners();
);

public static main : () ==> ()
main() ==
(
  IO`print("Executing all tests...\n");
  printTests();
);

end TestApp

```

Function or operation	Line	Coverage	Calls
getDesignersByRunway	111	100.0%	1
getEvent	73	100.0%	1
getFestival	41	100.0%	1
getFestivalEvents	48	100.0%	40
getFestivalEventsNames	63	100.0%	8
getFestivalUsers	54	100.0%	1
getFestivals	24	100.0%	1
getFestivalsNames	31	100.0%	1
getModelsByRunway	103	100.0%	1
getOneRunwayByEvent	96	100.0%	3
getRunwaysByEvent	80	100.0%	6
getRunwaysNames	86	100.0%	1
getUsers	17	100.0%	1
main	118	100.0%	1
printTests	10	100.0%	2
TestApp.vdmpp		100.0%	69

9 Tests

```

class Tests is subclass of MyTests

types
public String = seq of char;

instance variables
private static festivals: seq of FashionFestival := [];

```



```

private static events: seq of Event := [];
private static designers: seq of Designer := [];
private static models: seq of Model := [];
private static appUsers: set of FashionUser := {};
private static runways: seq of Runway := [];

```

operations

```

public run : () ==> ()
run() ==
(
  -- VARIABLE DECLARATIONS
  dcl f0: FashionFestival := new FashionFestival(" Porto Fashion Week", "04/05/2018", "
    10/05/2018", "Porto");
  dcl f1: FashionFestival := new FashionFestival(" Madrid Fashion Week", "26/08/2018", "
    30/08/2018", "Madrid");
  dcl ev0: Event := new Event(" Meet Portugal Fashion Designers", "04/05/2018", "Baixa", "12:00"
    , "3h", "Portugal", "Unisexo", "Primavera_Verao");
  dcl ev1: Event := new Event(" Fashion Night Out Porto", "10/06/2018", "Baixa", "20:00", "2h", "
    GeometricForms", "Unisexo", "Primavera_Verao");
  dcl ev2: Event := new Event(" Madrid Black Fashion", "12/06/2018", "Madrid", "19:00", "4h", "
    Fashion Sales", "Unisexo", "Outono_Inverno");
  dcl d0: Designer:= new Designer("Yves S. L.", "72", "Frances", "Paris", "Classico");
  dcl d1: Designer:= new Designer("Ralph Lauren", "69", "Frances", "Paris", "Classico");
  dcl m0: Model:= new Model("Sara Sampaio", "24", "Portuguesa", "New York");
  dcl m1: Model:= new Model("Claudia Schiffer", "47", "Alema", "Alemanha");
  dcl m2: Model:= new Model("Naomi Campbell", "47", "Inglesa", "Inglaterra");
  dcl m3: Model:= new Model("Kate Moss", "43", "Inglesa", "Inglaterra");
  dcl u0: FashionUser:= new FashionUser("Joao", "123456", "Joao", "30");
  dcl u1: FashionUser:= new FashionUser("Maria", "654321", "Maria", "34");
  dcl r0: Runway := new Runway(" Meet winter colleccion");
  dcl r1: Runway := new Runway(" African Power");
  dcl r2: Runway := new Runway(" Nautical Vibes");
  --festivals
  f0.insertEvent(ev0);
  f0.insertEvent(ev2);
  f1.insertEvent(ev0);
  f1.insertEvent(ev1);
  f0.insertFashionUser(u0);
  f0.insertFashionUser(u1);
  --events
  ev0.insertRunway(r0);
  ev1.insertRunway(r1);
  ev1.insertRunway(r2);
  ev2.insertRunway(r2);
  --runways
  r0.insertDesigner(d0);
  r0.insertDesigner(d1);
  r0.insertModel(m0);
  r1.insertDesigner(d1);
  r1.insertModel(m1);
  r1.insertModel(m2);
  r1.insertModel(m3);
  r2.insertDesigner(d0);
  r2.insertModel(m2);
  --USERS
  u0.insertEvent(ev0);
  u0.insertEvent(ev1);
  u0.insertDesigner(d0);
  u0.insertDesigner(d1);
  u0.insertModel(m0);
  u0.insertModel(m1);

```

```

    u1.insertEvent(ev2);
u1.insertEvent(ev0);
    u1.insertDesigner(d1);
u1.insertModel(m3);

festivals := festivals ^ [f0];
festivals := festivals ^ [f1];
events := events ^ [ev0];
events := events ^ [ev1];
events := events ^ [ev2];
designers := designers ^ [d0];
designers := designers ^ [d1];
models := models ^ [m0];
models := models ^ [m1];
models := models ^ [m2];
models := models ^ [m3];
appUsers := appUsers union {u0};
appUsers := appUsers union {u1};
runways := runways ^ [r0];
runways := runways ^ [r1];
runways := runways ^ [r2];

IO`print("(1/8) Test Creation of Elements\n");
assertNotNull(f0);
assertNotNull(f1);
assertNotNull(ev0);
assertNotNull(ev1);
assertNotNull(ev2);
assertNotNull(d0);
assertNotNull(d1);
assertNotNull(m0);
assertNotNull(m1);
assertNotNull(m2);
assertNotNull(m3);
assertNotNull(u0);
assertNotNull(u1);
IO`print("(1/8) Test Creation of Elements passed\n");

-- EXECUTE FASHION FESTIVAL
IO`print("(2/8) Test Fashion Festival\n");
assertEqual(" Porto Fashion Week", f0.getName());
assertEqual("04/05/2018", f0.getDateBegin());
assertEqual("10/05/2018", f0.getDateEnd());
assertEqual("Porto", f0.getLocal());
assertEqual(2, f0.getNumberEvents());
assertEqual(ev0.getName(), ((f0.getEvents() (2)).getName()));
assertEqual(2, f1.getNumberEvents());
assertEqual(ev1.getName(), ((f1.getEvents() (1)).getName()));
assertEqual(ev1.getName(), ((f1.getEvents() (1)).getName()));
assertEqual(2, card f0.getFashionUsers());
assertEqual(2, f0.getNumberFashionUsers());
assertEqual(("Name: " ^ f0.getName() ^ "\n" ^ "Date Begin: " ^ f0.getDateBegin() ^ "\n" ^ "Date End: " ^ f0
.getDateEnd() ^ "\n"
^ "Local: " ^ f0.getLocal() ^ "\n"), f0.printFashionFestival());
assertEqual(f0.getName() ^ "\n", f0.printFashionFestivalName());
IO`print("(2/8) Test Fashion Festival passed\n");

-- EXECUTE EVENT
IO`print("(3/8) Test Fashion Event\n");
assertEqual(" Meet Portugal Fashion Designers", ev0.getName());
assertEqual("04/05/2018", ev0.getDate());
assertEqual("Baixa", ev0.getLocal());
assertEqual("12:00", ev0.getTime());
assertEqual("3h", ev0.getDuration());
assertEqual("Portugal", ev0.getTheme());

```

```

    assertEquals("Unisexo", ev0.getGender());
    assertEquals("Primavera_Verao", ev0.getCollection());
    assertEquals(1, ev0.getNumberRunways());
    assertEquals(2, ev1.getNumberRunways());
    assertEquals(1, ev2.getNumberRunways());
    assertEquals(r0.getName(), (ev0.getRunways() (1)).getName());
    assertEquals(("Event Name: " + ev0.getName() + "\n" + "Date: " + ev0.getDate() + "\n" + "Time: " + ev0.getTime() + "\n" + "Theme: " + ev0.getTheme() + "\n"
        + "Gender: " + ev0.getGender() + "\n" + "Collection: " + ev0.getCollection() + "\n"), ev0.printEvent());
    IO.print("(3/8) Test Fashion Event passed\n");

    -- EXECUTE Runway
    IO.print("(4/8) Test Fashion Runway\n");
    assertEquals(r0.getDesignersNumber(), card r0.getDesigners());
    assertEquals(1, r0.getNumberModels());
    assertEquals(m0.getName(), (r0.getModels() (1)).getName());
    assertEquals((r0.getName() + "\n"), r0.printRunway());
    IO.print("(4/8) Test Fashion Runway passed\n");

    -- EXECUTE DESIGNER
    IO.print("(5/8) Test Fashion Designer\n");
    assertEquals("Yves S. L.", d0.getName());
    assertEquals("72", d0.getAge());
    assertEquals("Frances", d0.getNationality());
    assertEquals("Paris", d0.getAddress());
    assertEquals("Classico", d0.getStyle());
    assertEquals(("Designer Name: " + d0.getName() + "\n"
        + "Age: " + d0.getAge() + "\n"
        + "Nationality: " + d0.getNationality() + "\n"
        + "Address: " + d0.getAddress() + "\n"
        + "Style: " + d0.getStyle() + "\n"), d0.printDesigner());
    IO.print("(5/8) Test Fashion Designer passed\n");

    -- EXECUTE MODEL
    IO.print("(6/8) Test Fashion Model\n");
    assertEquals("Sara Sampaio", m0.getName());
    assertEquals("24", m0.getAge());
    assertEquals("Portuguesa", m0.getNationality());
    assertEquals("New York", m0.getAddress());
    assertEquals(("Model Name: " + m0.getName() + "\n"
        + "Age: " + m0.getAge() + "\n"
        + "Nationality: " + m0.getNationality() + "\n"
        + "Address: " + m0.getAddress() + "\n"), m0.printModel());
    IO.print("(6/8) Test Fashion Model passed\n");

    -- FASHION USER
    IO.print("(7/8) Test Fashion User\n");
    assertEquals("Joao", u0.getName());
    assertEquals("30", u0.getAge());
    assertEquals("Joao", u0.getUsername());
    assertEquals("123456", u0.getPassword());
    assertEquals(("Username: " + u0.getUsername() + "\n"
        + "Password: " + u0.getPassword() + "\n"
        + "Name: " + u0.getName() + "\n"
        + "Age: " + u0.getAge() + "\n"), u0.printUser());
    assertEquals(2, u0.getNumberEvents());
    assertEquals(ev0.getName(), (u0.getEvents() (1)).getName());
    assertEquals(ev1.getName(), (u0.getEvents() (2)).getName());
    assertEquals(2, u0.getNumberFavDesigners());
    assertEquals(d0.getName(), (u0.getDesigners() (1)).getName());
    assertEquals(d1.getName(), (u0.getDesigners() (2)).getName());

```

```

    assertEquals(2,u0.getNumberFavModels());
    assertEquals(m0.getName(), (u0.getModels() (1)).getName());
    assertEquals(m1.getName(), (u0.getModels() (2)).getName());
    IO`print("(7/8) Test Fashion User passed\n");

--TestApp
    IO`print("(8/8) Test Fashion App\n");
    assertEquals(2, card TestApp`getUsers());
    assertEquals(2, len TestApp`getFestivals());
    assertEquals(" Porto Fashion Week\n",TestApp`getFestivalsNames());
    assertEquals(f0.getName(), TestApp`getFestival(1).getName());
    assertEquals(2, len TestApp`getFestivalEvents(1));
    assertEquals(" Madrid Black Fashion",TestApp`getFestivalEventsNames(1));
    assertEquals(2, card TestApp`getFestivalUsers(1));
    assertEquals(ev2.getName(),TestApp`getEvent(1,1).getName());
    assertEquals(1,len TestApp`getRunwaysByEvent(1,1));
    assertEquals(" Nautical Vibes",TestApp`getRunwaysNames(1,1));
    assertEquals(r2.getName(),TestApp`getOneRunwayByEvent(1,1,1).getName());
    assertEquals(m2.getName(),(TestApp`getModelsByRunway(1,1,1) (1)).getName());
    assertEquals(1, card TestApp`getDesignersByRunway(1,1,1));
    assertEquals(2, len getFestivals());
    assertEquals(3, len getEvents());
    assertEquals(2, len getDesigners());
    assertEquals(4, len getModels());
    assertEquals(3, len getRunways());
    assertEquals(2, card getAppUsers());
    setAppUser("ariana","123456","ariana","21");
    assertEquals(3, card getAppUsers());
    IO`print("\n(8/8) Test Fashion App passed\n");

    IO`print("\nAll tests passed (8/8)\n");
);

public static getFestivals : () ==> seq of FashionFestival
getFestivals() == return festivals
post RESULT = festivals;

public static getEvents : () ==> seq of Event
getEvents() == return events
post RESULT = events;

public static getDesigners : () ==> seq of Designer
getDesigners() == return designers
post RESULT = designers;

public static getModels : () ==> seq of Model
getModels() == return models
post RESULT = models;

public static getAppUsers : () ==> set of FashionUser
getAppUsers() == return appUsers
post RESULT = appUsers;

public static getRunways : () ==> seq of Runway
getRunways() == return runways
post RESULT = runways;

```

```

public static setAppUser: (String) * (String) * (String) * (String) ==> ()
setAppUser(username,password,name,age) ==
(
  dcl u5 : FashionUser:= new FashionUser(username,password,name,age);
  appUsers := appUsers union {u5};
);

```

end Tests

Function or operation	Line	Coverage	Calls
getAppUsers	244	100.0%	3
getDesigners	236	100.0%	1
getEvents	232	100.0%	1
getFestivals	228	100.0%	16
getModels	240	100.0%	1
getRunways	248	100.0%	1
run	17	100.0%	1
setAppUser	252	100.0%	1
Tests.vdmpp		100.0%	25