

Fashion_Shows-MFES

December 28, 2017

Contents

1	Designer	1
2	Event	2
3	FashionFestival	4
4	FashionUser	6
5	Model	8
6	MyTestCase	9
7	TestApp	10
8	Tests	14

1 Designer

```
class Designer

types
  public String = seq of char;
  public DesignerName = String;
  public DesignerAge = nat;
  public DesignerNationality = String;
  public DesignerAddress = String;
  public DesignerStyle = String;

instance variables
  public name : DesignerName;
  public age : DesignerAge;
  public nationality : DesignerNationality;
  public address : DesignerAddress;
  public style : DesignerStyle;

operations

  public Designer :
    DesignerName *
    DesignerAge *
    DesignerNationality *
    DesignerAddress *
    DesignerStyle ==> 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;

public pure getAge : () ==> nat
  getAge() == return age;

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

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

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

end Designer

```

Function or operation	Line	Coverage	Calls
Designer	19	100.0%	2
getAddress	45	100.0%	1
getAge	39	100.0%	1
getName	36	100.0%	9
getNationality	42	100.0%	1
getStyle	48	100.0%	1
Designer.vdmpp		100.0%	15

2 Event

```

class Event

types
public String = seq of char;
public EventName = String;
public EventDate = String;
public EventLocal = String;
public EventTime = nat;
public EventDuration = nat;
public EventTheme = String;
public EventGender = <Homem> | <Mulher> | <Unisexo>;

```

```

    public EventCollection = <Outono_Inverno> | <Primavera_Verao>;

instance variables
public name : EventName := "";
public date : EventDate := "";
public local : EventLocal := "";
public time : EventTime := 0;
public duration : EventDuration := 0;
public theme : EventTheme := "";
public gender: EventGender := <Homem>;
public collection: EventCollection := <Outono_Inverno>;
--Lista de designers do evento
private designers: seq of Designer := [];
private numberDesigners: int := 0;
--Lista de modelos do evento
private models: seq of Model := [];
private numberModels: int := 0;

operations

public Event :
    EventName *
    EventDate *
    EventLocal *
    EventTime *
    EventDuration *
    EventTheme *
    EventGender *
    EventCollection ==> 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;

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

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

public pure getTime : () ==> nat
    getTime() == return time;

public pure getDuration : () ==> nat
    getDuration() == return duration;

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

```

```

public pure getGender : () ==> EventGender
  getGender() == return gender;

public pure getCollection : () ==> EventCollection
  getCollection() == return collection;

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

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

--DESIGNERS DO EVENTO
--Adiciona designer ao evento

public insertDesigner : Designer ==> ()
  insertDesigner(dg) ==
  (
    numberDesigners := numberDesigners + 1;
    designers := designers ^ [dg];
  );

--retorna nr designers do evento

public pure getNumberDesigners : () ==> int
  getNumberDesigners() == return numberDesigners;

--MODELOS DO EVENTO
--Adiciona model ao evento

public insertModel : Model ==> ()
  insertModel(md) ==
  (
    numberModels := numberModels + 1;
    models := models ^ [md];
  );

--retorna nr designers do evento

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

end Event

```

Function or operation	Line	Coverage	Calls
Event	31	100.0%	3
getCollection	74	100.0%	1
getDate	56	100.0%	1
getDesigners	80	100.0%	2
getDuration	65	100.0%	1
getGender	71	100.0%	1
getLocal	59	100.0%	1
getModels	77	100.0%	5

getName	53	100.0%	9
getNumberDesigners	93	100.0%	2
getNumberModels	106	100.0%	3
getTheme	68	100.0%	1
getTime	62	100.0%	1
insertDesigner	85	100.0%	2
insertModel	98	100.0%	5
Event.vdmpp		100.0%	38

3 FashionFestival

```

class FashionFestival

types
  public String = seq of char;
  public FestivalName = String;
  public FestivalDateBegin = String;
  public FestivalDateEnd = String;
  public FestivalLocal = String;

instance variables
  public name : FestivalName := "";
  public dateBegin : FestivalDateBegin := "";
  public dateEnd : FestivalDateEnd := "";
  public local : FestivalLocal := "";
  private events: seq of Event := [];
  private numberEvents: int := 0;

operations

  public FashionFestival :
    FestivalName *
    FestivalDateBegin *
    FestivalDateEnd *
    FestivalLocal
    ==> 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;

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

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

```

```

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

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

--Adiciona designer ao evento

public insertEvent : Event ==> ()
  insertEvent(ev) ==
  (
    events := [ev] ^ events;
    numberEvents := numberEvents + 1;
  );

--retorna nr designers do evento

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

end FashionFestival

```

Function or operation	Line	Coverage	Calls
FashionFestival	19	100.0%	2
getDateBegin	38	100.0%	1
getDateEnd	41	100.0%	1
getEvents	47	100.0%	2
getLocal	44	100.0%	1
getName	35	100.0%	1
getNumberEvents	59	100.0%	2
insertEvent	51	100.0%	3
FashionFestival.vdmpp		100.0%	13

4 FashionUser

```

class FashionUser

types
  public String = seq of char;
  public UserName = String;
  public UserAge = nat;
  public UserGender = <Homem> | <Mulher>;

instance variables
  public name: UserName;
  public age: UserAge;
  public gender: UserGender;
  --designers favoritos
  private designers: seq of Designer := [];
  private numberDesigners: int := 0;
  --eventos que utilizador vai
  private events: seq of Event := [];
  private numberEvents: int := 0;

```

operations

```
public FashionUser :
    UserName *
    UserAge *
    UserGender ==> FashionUser
FashionUser(nm, ag, gr) ==
(
    name := nm;
    age := ag;
    gender := gr;
    return self;
);

--retorna os parametros da class fashionUser

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

public pure getAge : () ==> nat
getAge() == return age;

public pure getGender : () ==> UserGender
getGender() == return gender;

--DESIGNERES FAVORITOS
--Adiciona designer favorito ao utilizador

public insertDesigner : Designer ==> ()
insertDesigner(d) ==
(
    numberDesigners := numberDesigners + 1;
    designers := designers ^ [d];
);

--retorna nr designers favoritos

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

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

--EVENTOS DO UTILIZADOR
--Adiciona evento

public insertEvent : Event ==> ()
insertEvent(ev) ==
(
    numberEvents := numberEvents + 1;
    events := events ^ [ev];
);

--Retorna nr designers favoritos

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

public getEvents: () ==> seq of Event
```

```

    getEvents() == return events;
end FashionUser

```

Function or operation	Line	Coverage	Calls
FashionUser	21	100.0%	1
getAge	37	100.0%	1
getDesigners	57	100.0%	2
getEvents	73	100.0%	2
getGender	40	100.0%	1
getName	34	100.0%	1
getNumberEvents	70	100.0%	1
getNumberFavDesigners	54	100.0%	1
insertDesigner	46	100.0%	2
insertEvent	62	100.0%	2
FashionUser.vdmpp		100.0%	14

5 Model

```

class Model

types
  public String = seq of char;
  public ModelName = String;
  public ModelAge = nat;
  public ModelNationality = String;
  public ModelAddress = String;

instance variables
  public name : ModelName;
  public age : ModelAge;
  public nationality : ModelNationality;
  public address : ModelAddress;

operations

  public Model :
    ModelName *
    ModelAge *
    ModelNationality *
    ModelAddress ==> 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;

```



```

public pure getAge : () ==> nat
  getAge() == return age;

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

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

end Model

```

Function or operation	Line	Coverage	Calls
Model	17	100.0%	4
getAddress	41	100.0%	1
getAge	35	100.0%	1
getName	32	100.0%	11
getNationality	38	100.0%	1
Model.vdmpp		100.0%	18

6 MyTestCase

```

class MyTestCase

operations

  -- Simulates assertion checking by reducing it to pre-condition checking.
  -- If 'arg' does not hold, a pre-condition violation will be signaled.
  -- Verification of pre-conditions must be enabled in order for this to work

  protected assertTrue : bool ==> ()
  assertTrue(arg) == return
  pre arg;

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

  protected assertFalse : bool ==> ()
  assertFalse(arg) == return
  pre not arg;

  -- 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 MyTestCase

```

Function or operation	Line	Coverage	Calls
assertEquals	28	38.8%	0
assertFalse	15	0.0%	0
assertNotNull	22	100.0%	13
assertTrue	8	0.0%	0
MyTestCase.vdmpp		48.3%	13

7 TestApp

```

class TestApp

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 users: seq of FashionUser := [];
private static eventsTemp: seq of Event := [];
private static modelsTemp: seq of Model := [];
private static designersTemp: seq of Designer := [];
private static festivalTemp: FashionFestival := new FashionFestival();

operations

public static printTests: () ==> ()
printTests() ==
(
  IO'print("\nExecuting Tests.vdmpp operations...\n");
  new Tests().run();
);

public static printFestivalsName: () ==> ()
printFestivalsName() ==
(

```

```

IO`print("\nPrint Festivals");
for counter = 1 to len Tests`getFestivals() do (
    IO`print("\n");
    IO`print(counter);
    IO`print((Tests`getFestivals() (counter)).getName());
    IO`print("\n");
    IO`print("\n");
);
);

public static getFestivalInf : (int) ==> ()
getFestivalInf(num) == (
    IO`print("\n");
    IO`print("Festival Name: ");
    IO`print((Tests`getFestivals() (num)).getName());
    IO`print("\n");
    IO`print("Date Begin: ");
    IO`print((Tests`getFestivals() (num)).getDateBegin());
    IO`print("\n");
    IO`print("Date End: ");
    IO`print((Tests`getFestivals() (num)).getDateEnd());
    IO`print("\n");
    IO`print("Local: ");
    IO`print((Tests`getFestivals() (num)).getLocal());
    IO`print("\n");
    IO`print("\n");
);

public static printEventsName: () ==> ()
printEventsName() ==
(
    IO`print("\nPrint Events");
    for counter = 1 to len Tests`getEvents() do (
        IO`print("\n");
        IO`print(counter);
        IO`print((Tests`getEvents() (counter)).getName());
        IO`print("\n");
        IO`print("\n");
    );
);

public static getEventInf: (int) * (int) ==> ()
getEventInf(optionFestiavl,optionEvent) ==
(
    IO`print("Event ");
    IO`print((getEventsByFestival2(optionFestiavl) (optionEvent)).getName());
    IO`print("\n");
    IO`print("Date: ");
    IO`print((getEventsByFestival2(optionFestiavl) (optionEvent)).getDate());
    IO`print("\n");
    IO`print("Local: ");
    IO`print((getEventsByFestival2(optionFestiavl) (optionEvent)).getLocal());
    IO`print("\n");
    IO`print("Time: ");
    IO`print((getEventsByFestival2(optionFestiavl) (optionEvent)).getTime());
    IO`print("h\n");
    IO`print("Duration: ");
    IO`print((getEventsByFestival2(optionFestiavl) (optionEvent)).getDuration());
    IO`print("h\n");
    IO`print("Theme: ");

```

```

    IO`print((getEventsByFestival2(optionFestival) (optionEvent)).getTheme());
    IO`print("\n");
    IO`print("Gender: ");
    IO`print((getEventsByFestival2(optionFestival) (optionEvent)).getGender());
    IO`print("\n");
    IO`print("Collection: ");
    IO`print((getEventsByFestival2(optionFestival) (optionEvent)).getCollection());
    IO`print("\n");
    IO`print("\n");
};

public static getEventsByFestival2 : (int) ==> seq of Event
getEventsByFestival2(num) == (
    for counter = 1 to len (Tests`getFestivals() (num)).getEvents() do (
        eventsTemp := eventsTemp ^ [(Tests`getFestivals() (num)).getEvents() (counter)];
    );
    return eventsTemp;
);

public static getEventsByFestival : (int) ==> ()
getEventsByFestival(num) == (
    IO`print("\n");
    for counter = 1 to len (Tests`getFestivals() (num)).getEvents() do (
        IO`print(counter);
        IO`print(": ");
        IO`print(((Tests`getFestivals() (num)).getEvents() (counter)).getName());
        IO`print("\n");
    );
);

public static getModelsNameByEvent : (int) * (int) ==> ()
getModelsNameByEvent(optionFestival,optionEvent) == (
    IO`print("\n");
    for counter = 1 to len (getEventsByFestival2(optionFestival) (optionEvent)).getModels() do (
        IO`print(counter);
        IO`print(": ");
        IO`print(((getEventsByFestival2(optionFestival) (optionEvent)).getModels() (counter)).
            getName());
        IO`print("\n");
    );
);

public static getModelsNameByEvent2 : (int) * (int) ==> seq of Model
getModelsNameByEvent2(optionFestival, optionEvent) == (
    festivalTemp := (Tests`getFestivals() (optionFestival));
    for counter = 1 to len (festivalTemp.getEvents() (optionEvent)).getModels() do (
        modelsTemp := modelsTemp ^ [(festivalTemp.getEvents() (optionEvent)).getModels() (counter)];
    );
    return modelsTemp;
);

public static getModelInf: (int) * (int) * (int) ==> ()
getModelInf(optionFestival, optionEvent,optionModel) ==
(
    festivalTemp := (Tests`getFestivals() (optionFestival));
    IO`print("Name: ");

```

```

IO`print((getModelsNameByEvent2(optionFestival,optionEvent) (optionModel)).getName());
IO`print("\n");
IO`print("Age: ");
IO`print((getModelsNameByEvent2(optionFestival,optionEvent) (optionModel)).getAge());
IO`print("\n");
IO`print("Nationality: ");
IO`print((getModelsNameByEvent2(optionFestival,optionEvent) (optionModel)).getNationality());
IO`print("\n");
IO`print("Address: ");
IO`print((getModelsNameByEvent2(optionFestival,optionEvent) (optionModel)).getAddress());
IO`print("\n");
IO`print("\n");

);

public static getDesginersNameByEvent2 :(int) * (int) ==> seq of Designer
getDesginersNameByEvent2(optionFestival, optionEvent) == (
    festivalTemp := (Tests`getFestivals() (optionFestival));
    for counter = 1 to len (festivalTemp.getEvents() (optionEvent)).getDesigners() do (
        designersTemp := designersTemp ^[((festivalTemp.getEvents() (optionEvent)).getDesigners()) (
            counter)];
    );
    return designersTemp;
);

public static getDesignersNameByEvent : (int) * (int) ==> ()
getDesignersNameByEvent(optionFestival,optionEvent) == (
    IO`print("\n");
    for counter = 1 to len (getEventsByFestival2(optionFestival) (optionEvent)).getDesigners() do
    (
        IO`print(counter);
        IO`print(": ");
        IO`print(((getEventsByFestival2(optionFestival) (optionEvent)).getDesigners() (counter)).
            getName());
        IO`print("\n");
    );
);

public static getDesignerInf: (int) * (int) * (int) ==> ()
getDesignerInf(optionFestival, optionEvent,optionDesigner) ==
(
    festivalTemp := (Tests`getFestivals() (optionFestival));
    IO`print("Name: ");
    IO`print((getDesginersNameByEvent2(optionFestival,optionEvent) (optionDesigner)).getName());
    IO`print("\n");
    IO`print("Age: ");
    IO`print((getDesginersNameByEvent2(optionFestival,optionEvent) (optionDesigner)).getAge());
    IO`print("\n");
    IO`print("Nationality: ");
    IO`print((getDesginersNameByEvent2(optionFestival,optionEvent) (optionDesigner)).
        getNationality());
    IO`print("\n");
    IO`print("Address: ");
    IO`print((getDesginersNameByEvent2(optionFestival,optionEvent) (optionDesigner)).getAddress());
    ;
    IO`print("\n");
    IO`print("Style: ");
    IO`print((getDesginersNameByEvent2(optionFestival,optionEvent) (optionDesigner)).getStyle());
    IO`print("\n");
    IO`print("\n");
);

```

```

public static main : () ==> ()
  main() ==
  (
    printTests();

  );

end TestApp

```

Function or operation	Line	Coverage	Calls
getDesginersNameByEvent2	166	0.0%	0
getDesignerInf	186	0.0%	0
getDesignersNameByEvent	175	0.0%	0
getEventInf	73	0.0%	0
getEventsByFestival	112	0.0%	0
getEventsByFestival2	104	0.0%	0
getFestivalInf	41	0.0%	0
getModelInf	146	0.0%	0
getModelsNameByEvent	125	0.0%	0
getModelsNameByEvent2	136	0.0%	0
main	210	100.0%	1
printEventsName	60	0.0%	0
printFestivalsName	27	0.0%	0
printTests	19	100.0%	1
TestApp.vdmpp		3.9%	2

8 Tests

```

class Tests is subclass of MyTestCase

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 users: seq of FashionUser := [];

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 Weekend", "26/08/2018", "30/08/2018", "Madrid");
dcl ev0: Event := new Event("BaixaShow", "04/05/2018", "Baixa", 12, 3,"flores", <Homem>, <Primavera_Verao>);
dcl ev1: Event := new Event("Fashion Night Out Porto", "10/06/2018","Baixa",20,2,"GeometricForms",<Unisexo>,<Primavera_Verao>);
dcl ev2: Event := new Event("Black Friday","12/06/2018","Vila do Conde",10,4,"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,"Alem","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", 30, <Homem>);

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];
users := users ^ [u0];

-- EXECUTE testInit()
IO`print("TestEvent.vdmpp (1/10): testInit() started...\n");
assertNotNull(f0);
assertNotNull(f1);
assertNotNull(ev0);
assertNotNull(ev1);
assertNotNull(ev2);
assertNotNull(d0);
assertNotNull(d1);
assertNotNull(ev2);
assertNotNull(m0);
assertNotNull(m1);
assertNotNull(m2);
assertNotNull(m3);
assertNotNull(u0);

-- EXECUTE Fashion Festival
-- EXECUTE testParams()
IO`print("TestFashionFestival.vdmpp (2/10): testParamsFestival() started...\n");
assertEqual("Porto Fashion Week", f0.getName());
assertEqual("04/05/2018", f0.getDateBegin());
assertEqual("10/05/2018", f0.getDateEnd());
assertEqual("Porto", f0.getLocal());

-- EXECUTE testInsertEventAtFestival()
IO`print("TestFashionFestival.vdmpp (3/10): testInsertEventAtFestival() started...\n");
f0.insertEvent(ev0);
f0.insertEvent(ev2);
f1.insertEvent(ev1);

assertEqual(2,f0.getNumberEvents());
assertEqual(ev0.getName(),((f0.getEvents() (2)).getName()));
assertEqual(1,f1.getNumberEvents());

```

```

assertEquals(ev1.getName(), ((fl.getEvents() (1)).getName()));

-- EXECUTE testEventParams()
IO`print("TestEvent.vdmpp (3/10): testEventParams() started...\n");
assertEquals("BaixaShow", ev0.getName());
assertEquals("04/05/2018", ev0.getDate());
assertEquals("Baixa", ev0.getLocal());
assertEquals(12, ev0.getTime());
assertEquals(3, ev0.getDuration());
assertEquals("flores", ev0.getTheme());
assertEquals(<Homem>, ev0.getGender());
assertEquals(<Primavera_Verao>, ev0.getCollection());

-- EXECUTE testInsertDesignerAtEvent()
IO`print("TestEvent.vdmpp (4/10): testInsertDesignerAtEvent() started...\n");
ev0.insertDesigner(d0);
ev1.insertDesigner(d1);
assertEquals(1, ev0.getNumberDesigners());
assertEquals(d0.getName(), (ev0.getDesigners() (1)).getName());
assertEquals(1, ev1.getNumberDesigners());
assertEquals(d1.getName(), (ev1.getDesigners() (1)).getName());

-- EXECUTE testInsertModelAtEvent()
IO`print("TestEvent.vdmpp (3/10): testInsertModelAtEvent() started...\n");
ev0.insertModel(m0);
ev0.insertModel(m1);
ev1.insertModel(m1);
ev1.insertModel(m2);
ev2.insertModel(m3);
assertEquals(2, ev0.getNumberModels());
assertEquals(m0.getName(), (ev0.getModels() (1)).getName());
assertEquals(m1.getName(), (ev0.getModels() (2)).getName());
assertEquals(2, ev1.getNumberModels());
assertEquals(m1.getName(), (ev1.getModels() (1)).getName());
assertEquals(m2.getName(), (ev1.getModels() (2)).getName());
assertEquals(1, ev2.getNumberModels());
assertEquals(m3.getName(), (ev2.getModels() (1)).getName());

--DESIGNER
-- EXECUTE testParams()
IO`print("TestDesigner.vdmpp (2/10): DesignerParams() started...\n");
assertEquals("Yves S. L.", d0.getName());
assertEquals(72, d0.getAge());
assertEquals("Frances", d0.getNationality());
assertEquals("Paris", d0.getAddress());
assertEquals("Classico", d0.getStyle());

--MODEL
-- EXECUTE testParams()
IO`print("TestModel.vdmpp (2/10): ModelParams() started...\n");
assertEquals("Sara Sampaio", m0.getName());
assertEquals(24, m0.getAge());
assertEquals("Portuguesa", m0.getNationality());
assertEquals("New York", m0.getAddress());

--USER
-- EXECUTE testParams()
IO`print("TestFashionUser.vdmpp (2/10): UserParams() started...\n");
assertEquals("Joao", u0.getName());
assertEquals(30, u0.getAge());
assertEquals(<Homem>, u0.getGender());

-- EXECUTE InsertFavoriteDesigner()
IO`print("TestFashionUser.vdmpp (2/10): InsertFavoriteDesigner() started...\n");
u0.insertDesigner(d0);

```



```

u0.insertDesigner(d1);
assertEqual(2,u0.getNumberFavDesigners());
assertEqual(d0.getName(), (u0.getDesigners() (1)).getName());
assertEqual(d1.getName(), (u0.getDesigners() (2)).getName());

-- EXECUTE UserEvents()
IO`print("TestFashionUser.vdmpp (2/10): UserEvents() started...\n");
u0.insertEvent(ev0);
u0.insertEvent(ev1);
assertEqual(2,u0.getNumberEvents());
assertEqual(ev0.getName(), (u0.getEvents() (1)).getName());
assertEqual(ev1.getName(), (u0.getEvents() (2)).getName());

-- PRINT SUCCESS MESSAGE
IO`print("TestFashionFestival.vdmpp (DONE): all tests successfully executed!\n");
);

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

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

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

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

public static getUsers : () ==> seq of FashionUser
getUsers() == return users;

end Tests

```

Function or operation	Line	Coverage	Calls
getDesigners	169	0.0%	0
getEvents	166	0.0%	0
getFestivals	163	0.0%	0
getModels	172	0.0%	0
getUsers	175	0.0%	0
run	16	100.0%	1
Tests.vdmpp		97.8%	1