

1. Tests

VisitWithDiscountTest

The screenshot shows an IDE with the `VisitWithDiscountTest` class open. The class contains three test methods: `testSummer()`, `testSummer2()`, and `testNOTSummer()`. The `Coverage` window displays the following data:

Element	Coverage	Covered Instruc	Missed Instru
Veterinario	9.8 %	123	1,126
test	7.5 %	57	700
src	13.4 %	66	426
veterinary	13.4 %	66	426
ClinicTest.java	0.0 %	0	251
Visit.java	15.7 %	16	86
Animal.java	0.0 %	0	62
VisitWithoutDiscount.java	0.0 %	0	15
FarmAnimal.java	0.0 %	0	6
Pet.java	0.0 %	0	6
VisitWithDiscount.java	100.0 %	50	0
VisitWithDiscount(String, ...)	100.0 %	9	0
calculateCost()	100.0 %	41	0

The JUnit window shows the test results: Finished after 0.021 seconds, Runs: 3/3, Errors: 0, Failures: 0.

The screenshot shows an IDE with the `VisitWithDiscount` class open. The class implements the `Visit` interface and includes a `calculateCost()` method. The `Coverage` window displays the following data:

Element	Coverage	Covered Instruc	Missed Instru
Veterinario	9.8 %	123	1,126
test	7.5 %	57	700
veterinary	7.5 %	57	700
ClinicTest.java	0.0 %	0	675
VisitWithoutDiscountTest.java	0.0 %	0	22
AllTests.java	0.0 %	0	3
VisitWithDiscountTest.java	100.0 %	57	0
testNOTSummer()	100.0 %	18	0
testSummer()	100.0 %	18	0
testSummer2()	100.0 %	18	0
src	13.4 %	66	426

The JUnit window shows the test results: Finished after 0.021 seconds, Runs: 3/3, Errors: 0, Failures: 0.

VisitWithoutDiscountTest

The screenshot shows an IDE with the following components:

- Editor:** Displays the implementation of `VisitWithoutDiscount` class, which extends `Visit`. It includes methods `super(date, shift)` and `calculateCost()`.
- Coverage:** A table showing coverage for `VisitWithoutDiscountTest` (Jan 19, 2021 9:30:17 PM).

Element	Coverage	Covered Instruc	Missed Instru
Veterinario	4.2 %	53	1,196
test	2.9 %	22	735
src	6.3 %	31	461
veterinary	6.3 %	31	461
Clinic.java	0.0 %	0	251
Visit.java	15.7 %	16	86
Animal.java	0.0 %	0	62
VisitWithDiscount.java	0.0 %	0	50
FarmAnimal.java	0.0 %	0	6
Pet.java	0.0 %	0	6
VisitWithoutDiscount.java	100.0 %	15	0
VisitWithoutDiscount(Str	100.0 %	5	0
calculateCost()	100.0 %	10	0

JUnit: Finished after 0.014 seconds. Runs: 1/1, Errors: 0, Failures: 0. The test runner shows `veterinary.VisitWithoutDiscountTest [Runner: JUnit 4]`.

The screenshot shows an IDE with the following components:

- Editor:** Displays the implementation of `VisitWithoutDiscountTest` class, which includes `@Before` and `@Test` methods.
- Coverage:** A table showing coverage for `VisitWithoutDiscountTest` (Jan 19, 2021 9:31:45 PM).

Element	Coverage	Covered Instruc	Missed Instru
Veterinario	4.2 %	53	1,196
test	2.9 %	22	735
src	6.3 %	31	461
veterinary	6.3 %	31	461
Clinic.java	0.0 %	0	251
Visit.java	15.7 %	16	86
Animal.java	0.0 %	0	62
VisitWithDiscount.java	0.0 %	0	50
FarmAnimal.java	0.0 %	0	6
Pet.java	0.0 %	0	6
VisitWithoutDiscount.java	100.0 %	15	0
VisitWithoutDiscount(Str	100.0 %	5	0
calculateCost()	100.0 %	10	0

JUnit: Finished after 0.015 seconds. Runs: 1/1, Errors: 0, Failures: 0. The test runner shows `veterinary.VisitWithoutDiscountTest [Runner: JUnit 4]`.

2. Code

Capturas de tests

The screenshot displays an IDE with two main panels. The left panel shows the source code for `ClinicTest.java`, which contains three test methods: `testVisitedAnimalsPets()`, `testVisitedAnimalsFarmAnimals()`, and `testVisitedAnimalsFutureDate()`. The right panel shows the 'Coverage' window for the test run.

Coverage Data:

Element	Coverage	Covered Instruc	Missed Instru
veterinary	80.5 %	396	96
Visit.java	52.0 %	53	49
Animal.java	40.3 %	25	37
Clinic.java	97.2 %	244	7
Clinic	97.2 %	244	7
bookVisit(Animal, boolean)	93.0 %	93	7
Clinic()	100.0 %	12	0
addVisit(Visit)	100.0 %	6	0
bookedDay(String, char)	100.0 %	27	0
freeDay(String)	100.0 %	19	0
profit(String)	100.0 %	32	0
visitedAnimals(String)	100.0 %	55	0
VisitWithDiscount.java	94.0 %	47	3
FarmAnimal.java	100.0 %	6	0
Pet.java	100.0 %	6	0
VisitWithoutDiscount.java	100.0 %	15	0
test	88.2 %	668	89

The test run summary shows: Finished after 0.037 seconds, Runs: 13/13, Errors: 0, Failures: 1. The failure trace indicates a `java.lang.AssertionError` in `testVisitedAnimalsFutureDate`.

This screenshot shows the same IDE environment, but with a different test method highlighted in the coverage window. The source code on the left is the same as in the first screenshot.

Coverage Data:

Element	Coverage	Covered Instruc	Missed Instru
veterinary	80.5 %	396	96
Visit.java	52.0 %	53	49
Animal.java	40.3 %	25	37
Clinic.java	97.2 %	244	7
Clinic	97.2 %	244	7
bookVisit(Animal, boolean)	93.0 %	93	7
Clinic()	100.0 %	12	0
addVisit(Visit)	100.0 %	6	0
bookedDay(String, char)	100.0 %	27	0
freeDay(String)	100.0 %	19	0
profit(String)	100.0 %	32	0
visitedAnimals(String)	100.0 %	55	0
VisitWithDiscount.java	94.0 %	47	3
FarmAnimal.java	100.0 %	6	0
Pet.java	100.0 %	6	0
VisitWithoutDiscount.java	100.0 %	15	0
test	88.2 %	668	89

The test run summary is identical: Finished after 0.037 seconds, Runs: 13/13, Errors: 0, Failures: 1. The failure trace now points to `testVisitedAnimalsFutureDate`.

Captura del código

The screenshot displays the source code of `Clinic.java` on the left and its coverage report on the right. The code includes a `visitedAnimals` method that returns a `HashSet<Animal>` based on the type of animal visited. The coverage report shows that the `visitedAnimals` method is covered 100% by the tests.

```
124 return profit;
125 }
126
127 /**
128  * Creates a set of the animals of a certain type that have been visited before
129  * today.
130  *
131  * @param type "pet" if we want to create a set with pets or "farm" otherwise
132  * @return a set of animals
133  */
134 public HashSet<Animal> visitedAnimals(final String type) {
135     HashSet<Animal> animales = new HashSet<Animal>();
136
137     if (type.equals("pet")) {
138         for (Visit visit : this.visits) {
139             Animal animal = visit.getAnimal();
140             if (animal instanceof Pet) {
141                 animales.add(animal);
142             }
143         }
144     } else { //Se supondrá que si no es pet será farm
145         for (Visit visit : this.visits) {
146             Animal animal = visit.getAnimal();
147             if (animal instanceof FarmAnimal) {
148                 animales.add(animal);
149             }
150         }
151     }
152
153     return animales;
154 }
```

Element	Coverage	Covered Instruc	Missed Instru
▼ ClinicTest.java	99.0 %	668	7
▼ ClinicTest	99.0 %	668	7
testVisitedAnimalsFuture	85.1 %	40	7
setUp()	100.0 %	6	0
test2AnimalesGranja()	100.0 %	57	0
testAnimalGranjaDespDo	100.0 %	57	0
testBookVisitFarmFirstVi	100.0 %	29	0
testBookVisitPetFirstVisi	100.0 %	29	0
testBookVisitPetSecondV	100.0 %	57	0
testBookVisitPetSecondV	100.0 %	50	0
testProfitFutureDates()	100.0 %	54	0
testProfitPastDates()	100.0 %	54	0
testVisitedAnimalsFarmA	100.0 %	59	0
testVisitedAnimalsFarmA	100.0 %	55	0
testVisitedAnimalsPetAni	100.0 %	55	0
testVisitedAnimalsPets()	100.0 %	59	0
▼ AllTests.java	0.0 %	0	3

JUnit 4 runner output:

```
Finished after 0.037 seconds
Runs: 13/13 Errors: 0 Failures: 1
veterinary.ClinicTest [Runner: JUnit 4] (0.024 s)
testVisitedAnimalsPets (0.005 s)
testVisitedAnimalsFutureDate (0.004 s)
testBookVisitFarmFirstVisit (0.011 s)
testProfitFutureDates (0.002 s)
testVisitedAnimalsFarmAnimals (0.000 s)
testVisitedAnimalsFarmAnimalsMixedVisits (0.000 s)
testBookVisitPetSecondVisitAfterFarm (0.000 s)
testAnimalGranjaDespDomestico (0.001 s)
testVisitedAnimalsPetAnimalsMixedVisits (0.000 s)
testProfitPastDates (0.000 s)
testBookVisitPetSecondVisitAfterPet (0.000 s)
```

3. Coverage

The screenshot displays the source code of `Clinic.java` on the left and its coverage report on the right. The code includes a `visitedAnimals` method that returns a `HashSet<Animal>` based on the type of animal visited. The coverage report shows that the `visitedAnimals` method is covered 100% by the tests.

```
148 Animal a1 = new FarmAnimal(1, "Blanca", "cow");
149 String tomorrow = LocalDate.now().plusDays(1).format(formatterDDMMYYYY);
150 assertEquals(tomorrow + "-D", c.bookVisit(a1, false));
151
152 String afterTomorrow = LocalDate.now().plusDays(2).format(formatterDDMMYYYY);
153 Animal a2 = new Pet(2, "Tara", "cat");
154 assertEquals(afterTomorrow + "-M", c.bookVisit(a2, false));
155 }
156
157 @Test
158 public final void test2AnimalesGranja() {
159     Animal a1 = new FarmAnimal(1, "Blanca", "cow");
160     String tomorrow = LocalDate.now().plusDays(1).format(formatterDDMMYYYY);
161     assertEquals(tomorrow + "-D", c.bookVisit(a1, false));
162
163     String afterTomorrow = LocalDate.now().plusDays(2).format(formatterDDMMYYYY);
164     Animal a2 = new FarmAnimal(2, "go", "Goat");
165     assertEquals(afterTomorrow + "-D", c.bookVisit(a2, false));
166 }
167
168 @Test
169 public final void testAnimalGranjaDespDomestico() {
170     Animal a1 = new Pet(2, "Tara", "cat");
171     String tomorrow = LocalDate.now().plusDays(1).format(formatterDDMMYYYY);
172     assertEquals(tomorrow + "-M", c.bookVisit(a1, false));
173
174     String afterTomorrow = LocalDate.now().plusDays(2).format(formatterDDMMYYYY);
175     Animal a2 = new FarmAnimal(2, "go", "Goat");
176     assertEquals(afterTomorrow + "-D", c.bookVisit(a2, false));
177 }
178
179 // -----
180 // Tests profit
181 // -----
182
183 @Test
184 public final void testProfitPastDates() {
185     Animal a1 = new FarmAnimal(1, "Blanca", "cow");
186     Visit v1 = new VisitWithoutDiscount("14/12/2018", "D");
187     v1.setGrossPrice(100);
188     v1.setAnimal(a1);
189     c.addVisit(v1);
190 }
```

Element	Coverage	Covered Instruc	Missed Instru
▼ veterinary	80.5 %	396	96
▼ Visit.java	52.0 %	53	49
▼ Animal.java	40.3 %	25	37
▼ Clinic.java	97.2 %	244	7
▼ Clinic	97.2 %	244	7
bookVisit(Animal, boolean)	93.0 %	93	7
Clinic()	100.0 %	12	0
addVisit(Visit)	100.0 %	6	0
bookedDay(String, char)	100.0 %	27	0
freeDay(String)	100.0 %	19	0
profit(String)	100.0 %	32	0
visitedAnimals(String)	100.0 %	55	0
VisitWithDiscount.java	94.0 %	47	3
FarmAnimal.java	100.0 %	6	0
Pet.java	100.0 %	6	0
VisitWithoutDiscount.java	100.0 %	15	0
test	88.2 %	668	89

JUnit 4 runner output:

```
Finished after 0.037 seconds
Runs: 13/13 Errors: 0 Failures: 1
veterinary.ClinicTest [Runner: JUnit 4] (0.024 s)
testVisitedAnimalsPets (0.005 s)
testVisitedAnimalsFutureDate (0.004 s)
testBookVisitFarmFirstVisit (0.011 s)
testProfitFutureDates (0.002 s)
testVisitedAnimalsFarmAnimals (0.000 s)
testVisitedAnimalsFarmAnimalsMixedVisits (0.000 s)
testBookVisitPetSecondVisitAfterFarm (0.000 s)
testAnimalGranjaDespDomestico (0.001 s)
testVisitedAnimalsPetAnimalsMixedVisits (0.000 s)
testProfitPastDates (0.000 s)
```

Clinic.java

ClinicTest.java

VisitWithDis...

VisitWithout...

VisitWithDis...

...

```

70  *
71  * @param animal an animal to visit.
72  * @param discount true if the visit has discount, false otherwise.
73  */
74  public String bookVisit(Animal a, boolean discount) {
75      boolean foundDate = false;
76      char shift = ' ';
77      LocalDate dtDate = LocalDate.now().plusDays(1);
78      String date = null;
79      while (!foundDate) {
80          date = dtDate.format(formatterDDMMYYYY);
81          if (a instanceof Pet) {
82              if (!bookedDay(date, 'D') && !bookedDay(date, 'M')) {
83                  shift = 'M';
84                  foundDate = true;
85              } else if (!bookedDay(date, 'D') && !bookedDay(date, 'A')) {
86                  shift = 'A';
87                  foundDate = true;
88              }
89          } else {
90              if (freeDay(date)) {
91                  shift = 'D';
92                  foundDate = true;
93              } else {
94                  // do nothing
95                  foundDate = false;
96              }
97          }
98          dtDate = dtDate.plusDays(1);
99      }
100      Visit v = null;
101      if (discount) {
102          v = new VisitWithDiscount(date, shift);
103      } else {
104          v = new VisitWithoutDiscount(date, shift);
105      }
106      v.setAnimal(a);
107      this.addVisit(v);
108      return date + "-" + shift;
109  }
110
111  /**
112   * Calculates the profit of the clinic until a date
113   *
114   * @param date a date.
115   * @return the profit.
116   */
117  public double profit(String date) {
118      double profit = 0;
119      for (Visit v : this.visits) {
120          if (LocalDate.parse(v.getDate(), formatterDDMMYYYY).isBefore(LocalDate.parse(date, formatterDDMMYYYY))) {
121              profit += v.getProfit();
122          }
123      }
124      return profit;
125  }

```

Coverage

ClinicTest (1) (Jan 19, 2021 9:36:33 PM)

Element	Coverage	Covered Instru	Missed Instru
▼ ClinicTest.java	99.0 %	668	7
▼ ClinicTest	99.0 %	668	7
testVisitedAnimalsFutureDate	85.1 %	40	7
setUp()	100.0 %	6	0
testZAnimalesGranja()	100.0 %	57	0
testAnimalGranjaDespDomestic	100.0 %	57	0
testBookVisitFarmFirstVisit	100.0 %	29	0
testBookVisitPetFirstVisit	100.0 %	29	0
testBookVisitPetSecondVisit	100.0 %	57	0
testBookVisitPetSecondVisitAfterFarm	100.0 %	50	0
testProfitFutureDates	100.0 %	54	0
testProfitPastDates	100.0 %	54	0
testVisitedAnimalsFarmAnimals	100.0 %	59	0
testVisitedAnimalsFarmAnimalsMixedVisits	100.0 %	55	0
testVisitedAnimalsPetAnimalsMixedVisits	100.0 %	55	0
testVisitedAnimalsPetAnimalsMixedVisitsAfterFarm	100.0 %	59	0
testZAnimalesGranja	100.0 %	59	0
▼ AllTests.java	0.0 %	0	3

JUnit

Finished after 0.037 seconds

Runs: 13/13 Errors: 0 Failures: 1

Failure Trace

testVisitedAnimalsFutureDate (0.004 s)
testBookVisitFarmFirstVisit (0.011 s)
testProfitFutureDates (0.002 s)
testVisitedAnimalsFarmAnimals (0.000 s)
testVisitedAnimalsFarmAnimalsMixedVisits (0.000 s)
testBookVisitPetSecondVisitAfterFarm (0.000 s)
testAnimalGranjaDespDomestic (0.001 s)
testVisitedAnimalsPetAnimalsMixedVisits (0.000 s)
testProfitPastDates (0.000 s)
testBookVisitPetSecondVisitAfterPet (0.000 s)
testBookVisitPetFirstVisit (0.000 s)
testZAnimalesGranja (0.001 s)

4. TestSuite

Clinic.java

ClinicTest.java

VisitWithDis...

AllTests.java

VisitWithDis...

...

```

1  package veterinary;
2
3  import org.junit.runner.RunWith;
4
5  @RunWith(Suite.class)
6  @SuiteClasses({ ClinicTest.class, VisitWithDiscountTest.class, VisitWithoutDiscountTest.class })
7  public class AllTests {
8
9  }

```

Coverage

AllTests (Jan 19, 2021 9:44:51 PM)

Element	Coverage	Covered Instru	Missed Instru
▼ Veterinario	91.8 %	1,146	103
▼ src	81.1 %	399	93
▼ test	98.7 %	747	10
▼ veterinary	98.7 %	747	10
▼ ClinicTest.java	99.0 %	668	7
▼ ClinicTest	99.0 %	668	7
▼ AllTests.java	0.0 %	0	3
▼ AllTests	0.0 %	0	3
▼ VisitWithDiscountTest.java	100.0 %	57	0
▼ VisitWithDiscountTest	100.0 %	57	0
▼ VisitWithoutDiscountTest.java	100.0 %	22	0
▼ VisitWithoutDiscountTest	100.0 %	22	0

JUnit

Finished after 0.04 seconds

Runs: 17/17 Errors: 0 Failures: 1

Failure Trace

testVisitedAnimalsFutureDate (0.005 s)
testBookVisitFarmFirstVisit (0.011 s)
testProfitFutureDates (0.001 s)
testVisitedAnimalsFarmAnimals (0.001 s)
testVisitedAnimalsFarmAnimalsMixedVisits (0.000 s)
testBookVisitPetSecondVisitAfterFarm (0.000 s)
testAnimalGranjaDespDomestic (0.000 s)
testVisitedAnimalsPetAnimalsMixedVisits (0.000 s)
testProfitPastDates (0.000 s)
testBookVisitPetSecondVisitAfterPet (0.001 s)
testBookVisitPetFirstVisit (0.000 s)