# Test Cases

## 4.1 Pilota

### Scelta delle Categorie:

**Parametro: NumeroPole**

|  |  |
| --- | --- |
| Parametro | Punteggio |
| Formato | (0<=punteggio<=575) |
| Categorie | Valore |
| VALORE[PUNTEGGIO\_VLR] | * 1. Valore in-Range [PUNTEGGIO\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: NumeroVittorie**

|  |  |
| --- | --- |
| Parametro | NumeroVittorie |
| Formato | 0<=numeroVittorie<=24 |
| Categorie | Valore |
| VALORE[NUMEROVITTORIE\_VLR] | * 1. Valore in-Range [NUMEROVITTORIE\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: NumeroPiazzamenti**

|  |  |
| --- | --- |
| Parametro | NumeroPiazzamenti |
| Formato | 0<=numeroPiazzamanti<=24 |
| Categorie | Valore |
| VALORE[NUMEROPIAZZAMENTI\_VLR] | * 1. Valore in-Range [NUMEROPIAZZAMENTI\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: NumeroRitiri**

|  |  |  |  |
| --- | --- | --- | --- |
| Parametro | | NumeroRitiri | |
| Formato | | 0<=numeroRitiri<=24 | |
| Categorie | | Valore | |
| VALORE[NUMERORITIRI\_VLR] | | * 1. Valore in-Range [NUMERORITIRI\_VLR\_OK]   2. Valore out of Range [error] | |
|  | **Test Case Name: TestGetPunteggio** | |  |
|  | Precondizioni: | |  |
|  | pilota.punteggio == 498 | |  |
|  | Flusso di Eventi: | |  |
|  | 1. Alla variabile exp viene assegnato il valore 498 2. Alla variabile act viene assegnato il valore di ritorno pilota.getPunteggio() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act | |  |
|  | Oracolo: | |  |
|  | assertEquals(exp,act) == true | |  |

|  |
| --- |
| **Test Case Name: TestSetPunteggio** |
| Precondizioni: |
| pilota.punteggio == 498 |
| Flusso di Eventi: |
| * + 1. Viene chiamato il metodo pilota.setPunteggio(500)     2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo pilota.getPunteggio() e 500 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestPunteggioRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| * + 1. Viene chiamato il metodo pilota.setPunteggio(-5000)     2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetNumeroVittorie** |
| Precondizioni: |
| pilota.numeroVittorie == 6 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 6 2. Alla variabile act viene assegnato il valore di ritorno pilota.getNumeroVittorie() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetNumeroVittorie** |
| Precondizioni: |
| pilota.numeroVittorie == 6 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo pilota.setNumeroVittorie(7) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo pilota.getNumeroVittorie() e 7 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestPunteggioRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo pilota.setNumeroVittorie(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene cenerata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetNumeroPole** |
| Precondizioni: |
| pilota.numeroPole == 7 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 7 2. Alla variabile act viene assegnato il valore di ritorno pilota.getNumeroPole() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetNumeroPole** |
| Precondizioni: |
| pilota.numeroPole == 7 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo pilota.setNumeroPole(7) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo pilota.getNumeroPole() e 7 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestNumeroPoleRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo pilota.setNumeroPole(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene cenerata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetNumeroPiazzamanti** |
| Precondizioni: |
| pilota.numeroPiazzamenti == 14 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 14 2. Alla variabile act viene assegnato il valore di ritorno pilota.getNumeroPiazzamanti() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetNumeroPiazzamenti** |
| Precondizioni: |
| pilota.numeroPiazzamanti == 14 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo pilota.setNumeroPiazzamanti(15) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo pilota.getNumeroPiazzamenti() e 15 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestNumeroPiazzamantiRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo pilota.setNumeroPiazzamenti(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetNumeroRitiri** |
| Precondizioni: |
| pilota.numeroRitiri == 0 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 0 2. Alla variabile act viene assegnato il valore di ritorno pilota.getNumeroRitiri() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetNumeroRitiri** |
| Precondizioni: |
| pilota.punteggio == 0 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo pilota.setNumeroRitiri(1) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo pilota.getNumeroRitiri() e 1 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestNumeroRitiriRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo pilota.setNumeroRitiri(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

## **5**.1 Setup

### Scelta delle Categorie:

**Parametro: CaricoAreodinamicoAnteriore**

|  |  |
| --- | --- |
| Parametro | CaricoAreodinamicoAnteriore |
| Formato | (0<=caricoAreodinamicoAnteriore<=10) |
| Categorie | Valore |
| VALORE[CARICOAREODINAMICOANTERIORE\_VLR] | * 1. Valore in-Range [CARICOAREODINAMICOANTERIORE\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: CaricoAreodinamicoPosteriore**

|  |  |
| --- | --- |
| Parametro | CaricoAreodinamicoPosteriore |
| Formato | 0<=caricoAreodinamicoPosteriore<=10 |
| Categorie | Valore |
| VALORE[CARICOAREODINAMICOPOSTERIORE\_VLR] | * 1. Valore in-Range [CARICOAREODINAMICOPOSTERIORE\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: CampanaturaAnteriore**

|  |  |
| --- | --- |
| Parametro | CampanaturaAnteriore |
| Formato | -5<=campanaturaAnteriore<=5 |
| Categorie | Valore |
| VALORE[CAMPANATURAANTERIORE\_VLR] | * 1. Valore in-Range [CAMPANATURAANTERIORE\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: CampanaturaPosteriore**

|  |  |
| --- | --- |
| Parametro | CampanaturaPosteriore |
| Formato | -5<=campanaturaPosteriore<=5 |
| Categorie | Valore |
| VALORE[CAMPANATURAPOSTERIORE\_VLR] | * 1. Valore in-Range [CAMPANATURAPOSTERIORE\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: PressioneFreni**

|  |  |
| --- | --- |
| Parametro | PressioneFreni |
| Formato | (0<=pressioneFreni<=100) |
| Categorie | Valore |
| VALORE[PRESSIONEFRENI\_VLR] | * 1. Valore in-Range [PRESSIONEFRENI\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: BarraAntirollioAnteriore**

|  |  |
| --- | --- |
| Parametro | BarraAntirollioAnteriore |
| Formato | 0<=barraAntirollioPosteriore<=10 |
| Categorie | Valore |
| VALORE[BARRAANTIROLLIOPOSTERIORE\_VLR] | * 1. Valore in-Range [BARRAATIROLLIOPOSTERIORE\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: BarraAntirollioPosteriore**

|  |  |
| --- | --- |
| Parametro | BarraAntirollioPosteriore |
| Formato | -5<=campanaturaAnteriore<=5 |
| Categorie | Valore |
| VALORE[CAMPANATURAANTERIORE\_VLR] | * 1. Valore in-Range [CAMPANATURAANTERIORE\_VLR\_OK]   2. Valore out of Range [error] |

**Parametro: CampanaturaPosteriore**

|  |  |
| --- | --- |
| Parametro | CampanaturaPosteriore |
| Formato | -5<=campanaturaPosteriore<=5 |
| Categorie | Valore |
| VALORE[CAMPANATURAPOSTERIORE\_VLR] | * 1. Valore in-Range [CAMPANATURAPOSTERIORE\_VLR\_OK]   2. Valore out of Range [error] |

|  |
| --- |
| **Test Case Name: TestGetDate** |
| Precondizioni: |
| setup.date == “10-10-2021” |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore “10-10-2021” 2. Alla variabile act viene assegnato il valore di ritorno setup.getDate() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetDate** |
| Precondizioni: |
| setup.date == “10-10-2021” |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setDate(“11-11-2021”) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getDate() e “11-11-2021” |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestGetId** |
| Precondizioni: |
| setup.Id == 1 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 1 2. Alla variabile act viene assegnato il valore di ritorno setup.getId() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetPunteggio** |
| Precondizioni: |
| setup.id == 1 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setId(2) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getId() e 2 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestGetCaricoAreodinamicoAnteriore** |
| Precondizioni: |
| setup.caricoAreodinamicoAnteriore == 5.3 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 5.3 2. Alla variabile act viene assegnato il valore di ritorno setup.getCaricoAreodinamicoAnteriore() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetcaricoAreodinamicoAnteriore** |
| Precondizioni: |
| setup.caricoAreodinamicoAnteriore == 5.3 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setCaricoAreodinamicoAnteriore(2) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getCaricoAreodinamicoAnteriore() e 2 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestPunteggioRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setCaricoAreodinamicoAnteriore(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetCaricoAreodinamicoPosteriore** |
| Precondizioni: |
| setup.caricoAreodinamicoPosteriore == 5.2 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 5.2 2. Alla variabile act viene assegnato il valore di ritorno setup.getCaricoAreodinamicoPosteriore() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetcaricoAreodinamicoPosteriore** |
| Precondizioni: |
| setup.caricoAreodinamicoPosteriore == 5.2 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setCaricoAreodinamicoPosteriore(2) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getCaricoAreodinamicoPosteriore() e 2 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestPunteggioRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setCaricoAreodinamicoPosteriore(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene cenerata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetCampanaturaAnteriore** |
| Precondizioni: |
| setup.campanaturaAnteriore == 1.5 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 1.5 2. Alla variabile act viene assegnato il valore di ritorno setup.getCampanaturaAnteriore() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetCampanaturaAnteriore** |
| Precondizioni: |
| setup.caricoAreodinamicoAnteriore == 1.5 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setCampanaturaAnteriore(1) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getCampanaturaAnteriore() e 1 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestCampanaturaAnterioreRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setCampanaturaAnteriore(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetCampanaturaPosteriore** |
| Precondizioni: |
| setup.campanaturaPosteriore == 1.3 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 1.3 2. Alla variabile act viene assegnato il valore di ritorno setup.getCampanaturaPosteriore() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetCampanaturaPosteriore** |
| Precondizioni: |
| setup.campanaturaPosteriore == 1.3 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setCampanaturaPosteriore (1) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getCampanaturaPosteriore() e 1 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestPunteggioRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setCampanaturaPosteriore(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene cenerata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetConvergenzaAnteriore** |
| Precondizioni: |
| setup.convergenzaAnteriore == 0.6 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 0.6 2. Alla variabile act viene assegnato il valore di ritorno setup.getConvergenzaAnteriore() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetConvergenzaAnteriore** |
| Precondizioni: |
| setup.convergenzaAnteriore == 0.6 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.set(0) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getConvergenzaAnteriore() e 0 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestConevergenzaAnterioreRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setConvergenzaAnteriore(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetConvergenzaPosteriore** |
| Precondizioni: |
| setup.convergenzaPosteriore == 0.3 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 0.3 2. Alla variabile act viene assegnato il valore di ritorno setup.getConvergenzaPosteriore() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetConvergenzaPosteriore** |
| Precondizioni: |
| setup.convergenzaPosteriore ==0.3 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setConvergenzaPosteriore(0) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getConvergenzaPosteriore() e 0 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestConvergenzaPosterioreRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setConvergenzaPosteriore(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene cenerata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetPressioneFreni** |
| Precondizioni: |
| setup.pressioneFreni == 50 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 50 2. Alla variabile act viene assegnato il valore di ritorno setup.getPressioneFreni() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetPressioneFreni** |
| Precondizioni: |
| setup.pressioneFreni == 50 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setPressioneFreni(15) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getPressioneFreni() e 15 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestPressioneFreni** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setPressioneFreni(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetBarraAntirollioAnteriore** |
| Precondizioni: |
| setup.barraAntirollioAnteriore == 5.8 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 5.8 2. Alla variabile act viene assegnato il valore di ritorno setup.getBarraAntirollioAnteriore() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetBarraAntirollioAnteriore** |
| Precondizioni: |
| setup.barraAntirollioAnteriore == 5.3 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setBarraAntirollioAnteriore(5) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getBarraAntirollioAnteriore() e 5 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestBarraAntirollioAnterioreRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setBarraAntirollioAnteriore(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestGetBarraAntirollioPosteriore** |
| Precondizioni: |
| setup.barraAntirollioPosterire == 5.5 |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il valore 5.5 2. Alla variabile act viene assegnato il valore di ritorno setup.getBarraAntirollioPosteriore() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetBarraAntirollioPosteriore** |
| Precondizioni: |
| setup.barraAntirollioPosteriore == 5.3 |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setCaricoAreodinamicoAnteriore(5.5) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo setup.getBarraAntirollioPosteriore() e 5.5 |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestBarraAntirollioPosterioreRange** |
| Precondizioni: |
|  |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo setup.setBarraAntirollioPosteriore(-5000) 2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

## 6.1 Tecnico

### Scelta delle Categorie:

### Parametro: SetupList

|  |  |
| --- | --- |
| Parametro | SetupList |
| Formato | setupList != null |
| Categorie | Formato |
| Formato [SETUPLIST\_FRM] | 1. Non rispetta il formato [error] 2. Rispetta il formato [SETUPLIST\_FRM\_OK] |

### Parametro: Setup

|  |  |
| --- | --- |
| Parametro | Setup |
| Formato | setup != null |
| Categorie | Formato |
| Formato [SETUP\_FRM] | 1. Non rispetta il formato [error] 2. Rispetta il formato [SETUP\_FRM\_OK] |

|  |
| --- |
| **Test Case Name: TestGetSetupList** |
| Precondizioni: |
| tecnico.setupList == […] |
| Flusso di Eventi: |
| 1. Alla variabile exp viene assegnato il riferimento a setupList 2. Alla variabile act viene assegnato il valore di ritorno tecnico.getSetupList() 3. Chiamo il metodo assertEquals passando come parametri le variabili exp e act |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetSetupList** |
| Precondizioni: |
| tecnico.setupList == […] |
| Flusso di Eventi: |
| 1. Vengono aggiunti due setup a setupList2 e successivamente viene chiamato tecnico.setSetupList(setupList2) 2. Chiamo il metodo assertEquals passando come parametri il valore di ritorno del metodo tecnico.getSetupList() e setupList2 3. Ripristiniamo con tecnico.setSetupList(setupList) |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestSetupListNull** |
| Precondizioni: |
| tecnico.setupList == […] |
| Flusso di Eventi: |
| 1. Viene chiamato il metodo tecnico.setSetupList (null) 2. Chiamo il metodo fail0; 3. Ripristino con tecnico.setSetupList(setupList) |
| Oracolo: |
| non viene generata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestAddSetup** |
| Precondizioni: |
| tecnico.setupList == […] |
| Flusso di Eventi: |
| * + 1. Creo un nuovo setup che chiamo setupToAdd     2. Chiamo tecnico.addSetup(setupToAdd)     3. Chiamo il metodo assertEquals passando come parametri true e tecnico.getSetupList().contains(setupToAdd) |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestRemoveSetup** |
| Precondizioni: |
| tecnico.setupList == […] |
| Flusso di Eventi: |
| * + 1. Creo un nuovo setup che chiamo setupToRemove     2. Chiamo tecnico.removeSetup(setupToAdd)     3. Chiamo il metodo assertEquals passando come parametri 0 e tecnico.getSetupList().contains(setupToRemove) |
| Oracolo: |
| assertEquals(exp,act) == true |

|  |
| --- |
| **Test Case Name: TestAddSetupNull** |
| Precondizioni: |
| tecnico.setupList == […] |
| Flusso di Eventi: |
| * + 1. Chiamo tecnico.addSetup(null)     2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |

|  |
| --- |
| **Test Case Name: TestRemoveSetupNull** |
| Precondizioni: |
| tecnico.setupList == […] |
| Flusso di Eventi: |
| * + 1. Chiamo tecnico.removeSetup(null)     2. Chiamo il metodo fail0; |
| Oracolo: |
| non viene generata alcuna eccezione |