**Nani Andrea cl.4^A int.**

**LISTA PRENOTAZIONI(PARTITA)**

**Tabelle di Test**

**PIANO DI TEST DELLA CLASSE: nodo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Precondizioni** | **Descrizione** | **Azioni** | **RisultatoAtteso** |
|  | Test costruttore Nodo | Costruire una prenotazione,  e passarla come parametro al costruttore Nodo.  Invocare getInfo() e  getLink() | getInfo=Prenotazione inserita  getLink=null |
|  | Test setInfo() | Costruire 2 Prenotazioni.  Il primo viene passato come parametro al costruttore di un nodo.  Il secondo viene utilizzato per verificare il getInfo | getInfo=p  getLink=null |
|  | Test setLink() | Costruire 2 Prenotazioni e 2 nodi.  Settare alla componente link del primo nodo il reference del secondo nodo | getInfo=p  getLink=p2 |

**PIANO DI TEST DELLA CLASSE: Prenotazione**

|  |  |  |  |
| --- | --- | --- | --- |
| **PRECONDIZIONI** | **DESCRIZIONE** | **AZIONI** | **RISULTATO ATTESO** |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*(); | Test Costruttore  Prenotazione | Costruire un oggetto di tipoPrenotazione con l'apposito costruttore e invocare i vari getter per verificare la costruzione dell'oggetto. | getCodiceIdentificativo=1  getCognome="Nani"  getNome="Andrea"  getData=data1  getTempoUtilizzo=1 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*(); | Test Costruttore di copia  Prenotazione | Costruire un oggetto di tipo Prenotazione con l'apposito costruttore(p1) e un altro oggetto Prenotazione(p2) con l'apposito costruttore di copia e invocare i vari getter per verificare l'oggetto di copia si uguale alla prima prenotazione | P2.getCodiceIdentificativo=p1. getCodiceIdentificativo  P2.getCognome=p1. getCognome  P2.getNome= p1.getNome  P2.getData= p1.getData  P2.getTempoUtilizzo= p1.getTempoUtilizzo |
|  | Test Costruttore default  Prenotazione | Costruire un oggetto di tipo Prenotazione utilizzando il costruttore di default e invocare i vari getter per verificare la costruzione dell'oggetto | getCodiceIdentificativo=0  getCognome=""  getNome=""  getData=null  getTempoUtilizzo=0 |
|  | Test setCodiceIdentificativo () | Costruire un oggetto di tipo Prenotazione utilizzando il costruttore di default e Invocare il metodo setCodiceIdentificativo (1) e dopo il metodo getCodiceIdentificativo()per verificare il corretto funzionamento del metodo setter | getCodiceIdentificativo =1 |
|  | Test setCognome () | Costruire un oggetto di tipo Prenotazione utilizzando il costruttore di default e Invocare il metodo setCognome("nani") e dopo il metodo getCognome() per verificare il corretto funzionamento del metodo setter | getCognome="nani" |
|  | Test setNome () | Costruire un oggetto di tipo Prenotazione utilizzando il costruttore di default e Invocare il metodo setNome("Andrea") e dopo il metodo getNome()per verificare il corretto funzionamento del metodo setter | getNome="Andrea " |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*(); | Test setDataInizio () | Costruire un oggetto di tipo Prenotazione utilizzando il costruttore di default e Invocare il metodo setDataInizio (data1)e dopo il metodo getDataInizio()per verificare il corretto funzionamento del metodo setter | getDataOra= data1 |
|  | Test setTempoUtilizzo() | Costruire un oggetto di tipo Prenotazione utilizzando il costruttore di default e Invocare il metodo setTempoUtilizzo(1)e dopo il metodo getTempoUtilizzo()per verificare il corretto funzionamento del metodo setter | getTempoUtilizzo=1 |
|  | Test setTariffaCampo () | Costruire un oggetto di tipo Prenotazione utilizzando il costruttore di default e Invocare il metodo setTariffaCampo(1)e dopo il metodo getTariffaCampo()per verificare il corretto funzionamento del metodo setter | getTariffaCampo=1 |
|  | Test setTariffaDocce () | Costruire un oggetto di tipo Prenotazione utilizzando il costruttore di default e Invocare il metodo setTariffaDocce(1)e dopo il metodo getTariffaDocce()per verificare il corretto funzionamento del metodo setter | getTariffaDocce=1 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E una prenotazione(p) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1); | Test ToString() | Richiamare il metodo toString() sulla prenotazione appena creata e verificare che la stringa generata con il toString() sia uguale a quella attesa con il metodo compareTo() | "codice=1 cognome=nani  nome=andrea  data 1/1/2018" |

**PIANO DI TEST DELLA CLASSE: Partita(Lista di Prenotazioni)**

|  |  |  |  |
| --- | --- | --- | --- |
| **PRECONDIZIONI** | **DESCRIZIONE** | **AZIONI** | **RISULTATO ATTESO** |
|  | Test costruttore  Partita() | Costruire un oggetto di tipo Partita e invocare il metodo getElementi() per verificare che la lista sia vuota | getElementi=0 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E una prenotazione(p) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1); | Test Inserisci Prenotazione() | Costruire un oggetto di tipo Partita con l'apposito costruttore e provare a inserire la prenotazione p generata precedentemente(vedi precondizioni) con il metodo inserisciPrenotazione(p) e successivamente verificare che sia stata inserita correttamente verificando il numero di elementi con getElementi() | getElementi=1 |
|  | Test toString() vuoto | Costruire un oggetto di tipo Partita con l'apposito costruttore e invocare il metodo toString sulla partita costruita(che è vuota ovvero priva di prenotazioni).  Verificare che venga comunicato il messaggio atteso | "elenco prenotazioni palestra: nessuna prenotazione"  getElementi=0 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E una prenotazione(p) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1); | Test toString() | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire la prenotazione p generata precedentemente(vedi precondizioni) con il metodo inserisciPrenotazione(p).  invocare il metodo toString sulla partita costruita.  Verificare che venga comunicato il messaggio atteso | "elenco prenotazioni palestra: \n" +  "codice=1 cognome=nani  nome=andrea  data 1/1/2018"  getElementi=1 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E una prenotazione(p) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1); | Test inserisciInTesta  () | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire la prenotazione p generata precedentemente(vedi precondizioni) con il metodo inserisciInTesta (p).  Verificare il corretto inserimento della Prenotazione nel primo posto della lista con il metodo toString() | "elenco prenotazioni palestra:codice=1 cognome=nani  nome=andrea  data 1/1/2018"  getElementi=1 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data1,1); | Test inserisciInCoda  () | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire la prenotazione p generata precedentemente(vedi precondizioni) con il metodo inserisciInCoda (p).  Verificare il corretto inserimento della Prenotazione nel ultimo posto della lista con il metodo toString() | "elenco prenotazioni palestra:--codice=1 cognome=nani  nome=andrea  data 1/1/2018  ----"codice= 2  cognome=a  nome=a  data 1/1/2018"  "  getElementi=2 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data1,1); | Test eliminaInTesta  () | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire le 2 prenotazione p generate precedentemente(vedi precondizioni).  Eliminare la prima prenotazione della lista invocando il metodo eliminaInTesta.  Verificare la corretta eliminazione con il metodo toString() | "elenco prenotazioni palestra:--codice= 2  cognome=a  nome=a  data 1/1/2018"  getElementi=1 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data1,1); | Test eliminaInCoda  () | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire le 2 prenotazione p generate precedentemente(vedi precondizioni).  Eliminare l'ultima prenotazione della lista invocando il metodo eliminaInCoda.  Verificare la corretta eliminazione con il metodo toString() | "elenco prenotazioni palestra:codice=1 cognome=nani  nome=andrea  data 1/1/2018"  getElementi=1 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E una prenotazione(p) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1); | Test CalcolaPrezzo() | Costruire un oggetto di tipo Partita con l'apposito costruttore e provare a inserire la prenotazione p generata precedentemente(vedi precondizioni) con il metodo inserisciPrenotazione(p).  Calcolare il prezzo di questa Prenotazione con l'apposito metodo calcolaPrezzo() e verificare che il prezzo sia stato calcolato correttamente | calcolaPrezzo(p)==35 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E una prenotazione(p) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1); | test RegistraPrenotazione() | Costruire un oggetto di tipo Partita con l'apposito costruttore e provare a inserire la prenotazione p generata precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione() che verificherà i vari attributi della prenotazione.  verificare che sia andato tutto a buon fine con il metodo tostring() | "elenco prenotazioni palestra:codice=1 cognome=nani  nome=andrea  data 1/1/2018"  getElementi=1 |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore e fare in modo che p2 sia la copia di p1 per avere la stessa data | test RegistraPrenotazioneStessaData() | Costruire un oggetto di tipo Partita con l'apposito costruttore e provare a inserire le prenotazioni p1 e p2 generate precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione().  Verificare che si esegua l'eccezione di tipo PartitaException | expected=partitaException.**class** |
| Creare un oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E una prenotazione(p) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1); | test EliminaPrenotazione() | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire la prenotazione p generata precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione() e successivamente eliminarla dalla lista con il metodo eliminaPrenotazione().  Verificare la corretta eliminazione con il metodo toString() | "elenco prenotazioni palestra: nessuna prenotazione"  getElementi=0 |
| Creare due oggetti di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data2,1); | TestEliminaPrenotazioneCodiceInesistente() | Costruire un oggetto di tipo Partita con l'apposito costruttore e provare a inserire le prenotazioni p1 e p2 generate precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione().  Provare ad eliminare una prenotazione con il metodo eliminaPrenotazione(). ma inserendo il codice errato.  Verificare che sia stata sollevata un eccezione di tipo PartitaException | expected=partitaException.**class** |
| Creare un’oggetto di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E una prenotazione(p) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1); | test finepartita() | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire la prenotazione p generata precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione() e successivamente registrare la fine della prenotazione.  Verificare il corretto funzionamento della funzione | Getelementi=0 |
| Creare due oggetti di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data2,1); | test finepartitaConErrore() | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire le prenotazioni p1,p2 generate precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione() e successivamente registrare la fine della prenotazione ma inserendo un codice erratto.  Verificare che sia stata sollevata un eccezione di tipo PartitaException | expected=partitaException.**class** |
| Creare due oggetti di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data2,1); | TestselectionSortCrescenteData() | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire le prenotazioni p1,p2 generate precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione().  Trasformare la lista partita in un array e ordinarlo in ordine di data tramite l'apposita funzione selectionSortCrescenteData().  Verificare il corretto ordinamento | Array ordinato per data |
| Creare due oggetti di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data2,1); | TestselectionSortCrescenteNome() | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire le prenotazioni p1,p2 generate precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione().  Trasformare la lista partita in un array e ordinarlo in ordine di nome tramite l'apposita funzione selectionSortCrescenteNome().  Verificare il corretto ordinamento | Array ordinato per nome |
| Creare due oggetti di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data2,1); | test modificaTariffePalestra() | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire le prenotazioni p1,p2 generate precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione().  Modificare le tariffe delle prenotazioni con l'apposito metodo ModificaTariffePalestra().  Calcolare il prezzo e vedere la corretta riuscita della modifica | p.*getTariffaore*()==1  &&  p.*getTariffadocce*()==1 |
| Creare due oggetti di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data2,1); | test  VerificaDisponibiltàPalestra() | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire le prenotazioni p1,p2 generate precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione().  Verificare la disponibilità della palestra con l'apposito metodo verificaDisponibilità(). | "PALESTRA LIBERA" |
| Creare due oggetti di tipo LocalDateTime.(una data)  Es. LocalDateTime data1=LocalDateTime.*now*()  E 2 prenotazioni(p1,p2) con l'apposito costruttore  Es. Prenotazione p=**new** Prenotazione(1,"nani","andrea",data1,1);  Prenotazione p2=**new** Prenotazione(2,"a","a",data2,1); | test  VerificaDisponibiltàPalestraConErrore() | Costruire un oggetto di tipo Partita con l'apposito costruttore e inserire le prenotazioni p1,p2 generate precedentemente(vedi precondizioni) con il metodo RegistraPrenotazione().  Verificare la disponibilità della palestra con l'apposito metodo verificaDisponibilità() in una data in cui la palestra è occupata.  Verificare che sia stata sollevata un eccezione di tipo PartitaException | expected=partitaException.**class** |

**PIANO TEST DI INTEGRAZIONE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N** | **PRECONDIZIONI** | **AZIONI** | **RISULTATO ATTESO** | **REQUISITI VERIFICATI** | **ERRORI** |
| 1 | Azione 0 | Avviare l’esecuzione | Visualizzazione del menù | Deserializzazione oggetti da file binario  chiamato "prenotazioni.bin" |  |
| 2 |  | Digitare “Z” | Visualizzare “Formato inserito non corretto...REINSERIRE” |  |  |
| 3 |  | Digitare 10 | Visualizzare “Opzione non disponibile, reinserire” |  |  |
| 4 |  | Digitare 0 “ESCI” | Visualizzare “ARRIVEDERCI” |  |  |
| 5 | Azione 1 | Digitare 1 “inserire nuova prenotazione” | Inserire i vari dati di una prenotazione | Requisito 1 |  |
| 6 |  | Inserire il codice identificativo della prenotazione: a | Visualizzare “formato inserito errato..RIPROVARE L'OPERAZIONE” |  |  |
| 7 |  | Inserire la data: 56/20/2018 | Visualizzare “data inserita errata..RIPROVARE L'OPERAZIONE” |  |  |
| 8 |  | Inserire l'ora :  45 | Visualizzare “data inserita errata..RIPROVARE L'OPERAZIONE” |  |  |
| 9 |  | Inserire i minuti:  456 | Visualizzare “data inserita errata..RIPROVARE L'OPERAZIONE” |  |  |
| 10 |  | Inserire ore di utilizzo:  z | Visualizzare ” formato inserito errato..RIPROVARE L'OPERAZIONE” |  |  |
| 11 |  | Inserire data:  data già in uso da una prenotazione | Visualizzare" Palestra già occupata" | Requisito 1:verificare che la data sia libera |  |
| 12 |  | Inserire il codice identificativo della prenotazione: -1 | Visualizzare ""codice inserito errato..Reinserire"" |  |  |
|  |  | Inserire il tempo di utilizzo della prenotazione: -1 | Visualizzare ""valore inserito errato..Reinserire"" |  |  |
| 13 |  | Inserire dati di una prenotazione correttamente | Visualizzare" Prenotazione effettuata" | Serializzazione della prenotazione su un file binario chiamato "prenotazioni.bin" |  |
| 14 | Azione 2 | Digitare 2:"elimina prenotazione" | Visualizzare"Inserire codice della prenotazione da eliminare" | Requisito 2 |  |
| 15 |  | Digitare 2:"elimina prenotazione" ma la lista è vuota | Visualizzare" Nessuna prenotazione presente" |  |  |
| 16 |  | Inserire codice della prenotazione da eliminare:  e | Visualizzare ” formato inserito errato..RIPROVARE L'OPERAZIONE” |  |  |
| 17 |  | Inserire codice della prenotazione da eliminare:  56 | Visualizzare" Nessuna prenotazione con questo codice identificativo" |  |  |
| 18 |  | Inserire codice della prenotazione da eliminare:  -1 | Visualizzare" codice inserito errato..Reinserire" |  |  |
| 19 |  | Inserire codice della prenotazione da eliminare:  1 | Visualizzare" eliminazione avvenuta con successo" | Eliminazione dal file binario della prenotazione che è stata eliminata |  |
| 20 | Azione 3 | Digitare 3:"registra fine partita" | Visualizzare" inserisci il codice della tua prenotazione" | Requisito 3 |  |
| 21 |  | Inserire codice della prenotazione da eliminare:  e | Visualizzare ” formato inserito errato..RIPROVARE L'OPERAZIONE” |  |  |
| 22 |  | Inserire codice della prenotazione da eliminare:  45 | Visualizzare" Nessuna prenotazione con questo codice identificativo" |  |  |
| 23 |  | Inserire codice della prenotazione da eliminare:  -1 | Visualizzare" codice inserito errato..Reinserire" |  |  |
| 24 |  | Inserire codice della prenotazione da eliminare:  1 | Visualizzare"  il prezzo è: N€  scritture effettuata correttamente  eliminazione avvenuta con successo" | Requisito 3:Eliminazione dal file binario della prenotazione finita e salvataggio dello scontrino con ora inizio e ora fine della prenotazione e pagamento su un file di testo |  |
| 25 |  | Inserire codice della prenotazione da eliminare:  ma nessuna prenotazione presente nella lista | Visualizzare" Nessuna prenotazione presente" |  |  |
| 26 | Azione 4 | Digitare 4:visualizzare elenco prenotazioni in base al nome | Visualizzare l'elenco delle prenotazioni in base al nome | requisito 5 |  |
|  |  | Digitare 4:visualizzare elenco prenotazioni in base al nome mala lista è vuota | Visualizzare" Nessuna prenotazione presente" |  |  |
| 27 | Azione 5 | Digitare 5:visualizzare elenco prenotazioni in base alla data | Visualizzare l'elenco delle prenotazioni in base alla data | requisito 4 |  |
| 28 |  | Digitare 4:visualizzare elenco prenotazioni in base al nome mala lista è vuota | Visualizzare" Nessuna prenotazione presente" |  |  |
| 29 | Azione 6 | Digitare 6:modifica tariffe palestra | Visualizzare"inserire nuova tariffa campo\doccia" | Requisito 6 |  |
| 30 |  | Digitare 6:modifica tariffe palestra lista è vuota | Visualizzare" Nessuna prenotazione presente" |  |  |
| 31 |  | Inserire nuova tariffa campo:  -1 | Visualizzare" codice inserito errato..Reinserire" |  |  |
| 32 |  | Inserire nuova tariffa campo:  a | Visualizzare ” formato inserito errato..RIPROVARE L'OPERAZIONE” |  |  |
| 33 |  | Inserire nuova tariffa docce:  -1 | Visualizzare" codice inserito errato..Reinserire" |  |  |
| 34 |  | Inserire nuova tariffa docce:  a | Visualizzare ” formato inserito errato..RIPROVARE L'OPERAZIONE” |  |  |
| 35 |  | Inserire nuova tariffa campo:  1  Inserire nuova tariffa docce:  1 | Visualizzare" Modifica avvenuta con successo" |  |  |
| 36 | Azione 7 | Digitare 7:Verificare la disponibilità del campo | Visualizzare"inserire il mese nel quale si desidera verificare la disponibilità: " | Requisito 7 |  |
| 37 |  | Digitare 7:Verificare la disponibilità del campo.  Ma la lista è vuota | Visualizzare" Palestra libera in tutte le date" |  |  |
| 38 |  | Inserire la data: 56/20/2018 | Visualizzare “data inserita errata..RIPROVARE L'OPERAZIONE” |  |  |
| 39 |  | Inserire l'ora :  45 | Visualizzare “data inserita errata..RIPROVARE L'OPERAZIONE” |  |  |
| 40 |  | Inserire i minuti:  456 | Visualizzare “data inserita errata..RIPROVARE L'OPERAZIONE” |  |  |
| 41 |  | Inserire la data: data in cui la palestra è occupata | Visualizzare ”PALESTRA OCCUPATA” |  |  |
| 42 |  | Inserire la data:  data in cui la palestra è libera | Visualizzare"PALESTRA LIBERA" |  |  |
| 43 |  | Digitare 0:esci | Visualizzare"ARRIVEDERCI" |  |  |

**DIAGRAMMA DELLE CLASSI**

**PALESTRA COMUNALE**

