Fondamenti di Informatica II
Progettazione del Software

Esercitazione 5 – Simulazione d’esame

25/05/2018

Requisiti

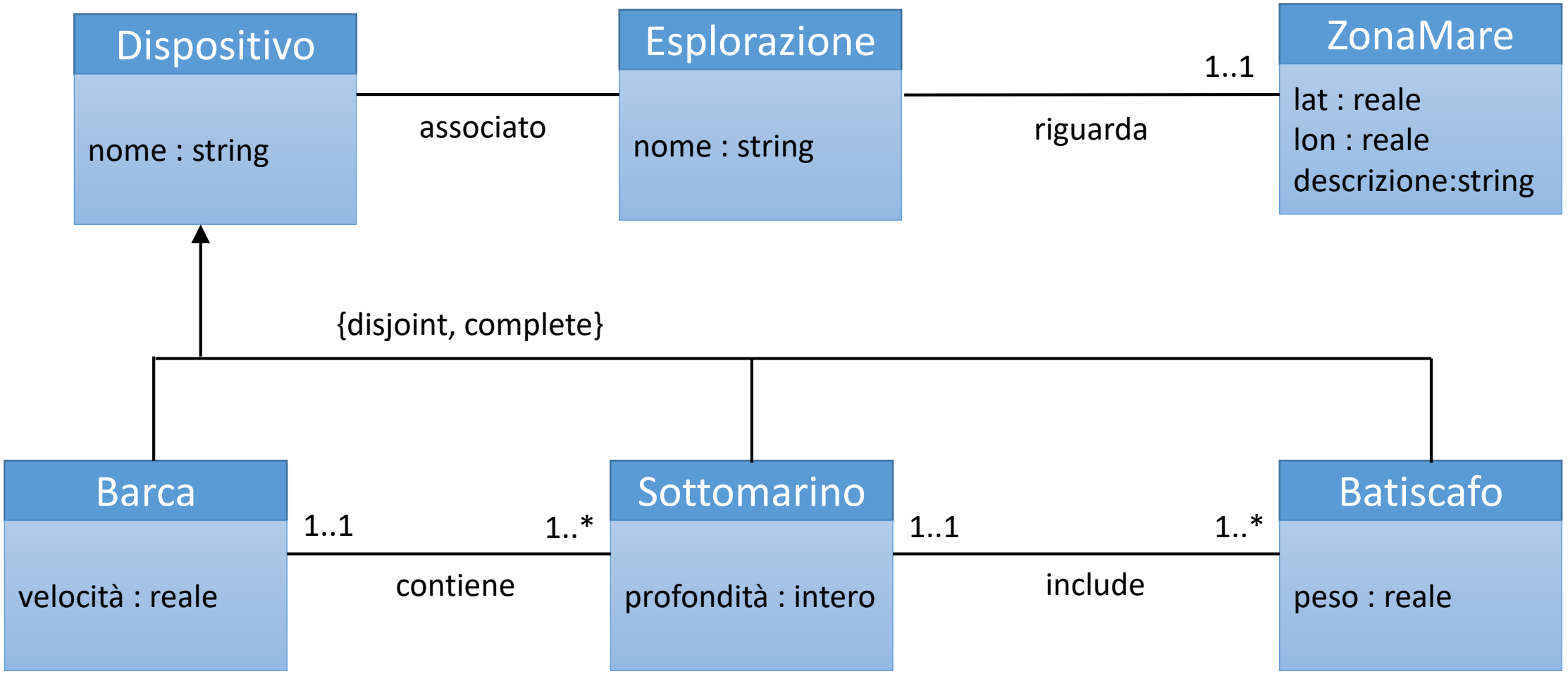
Si legga il file *esame.pdf* presente nella cartella dell'esercitazione.

Esercizi

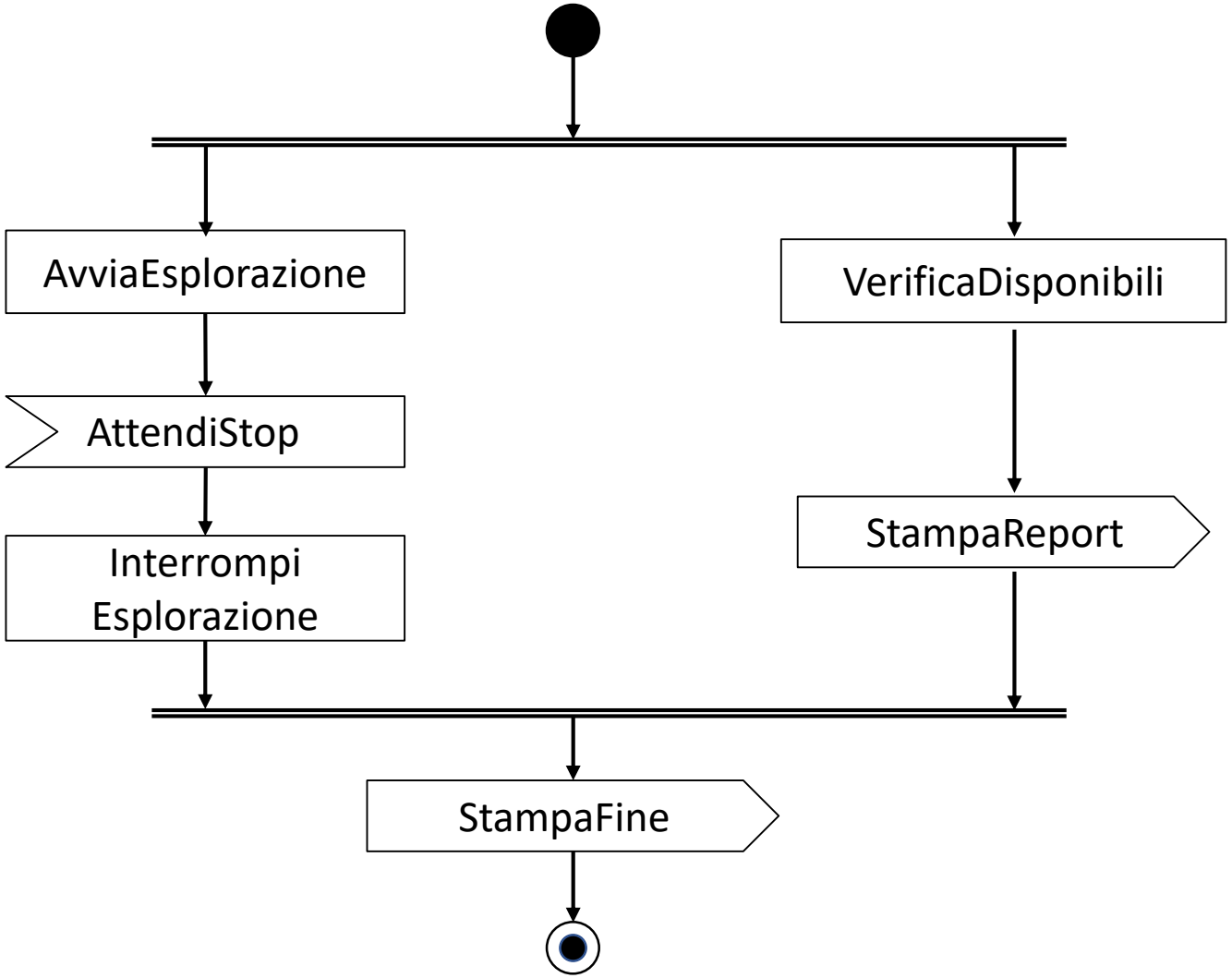
Domanda 1. Basandosi sui requisiti riportati sopra, effettuare la fase di analisi producendo lo schema concettuale, comprensivo di diagramma stati e transizioni, diagramma delle attività e specifica degli stati, delle transizioni e di tutte le attività, atomiche e non.

Domanda 2. Effettuare la realizzazione, estendendo il programma JAVA nella cartella *java-aux-250518* e motivando, qualora ce ne fosse bisogno, le scelte effettuate. **Si consiglia** di iniziare con l'implementazione dei dettagli necessari per gestire lo stato e le transizioni della classe *Sottomarino* (con classe *SottomarinoFired*).

Fase di Analisi – Diagramma delle Classi



Fase di Analisi – Diagramma delle Attività



Specifica attività I/O

InizioSpecificaAttivitàAtomica AttendiStop

AttendiStop(e : Esplorazione) : ( )

pre: -

post: Attende che l'utente decida di terminare l'esplorazione e.

FineSpecificaAttivitàAtomica

Specifica attività I/O (2)

InizioSpecificaAttivitàAtomica StampaReport

StampaReport(c : Insieme(Barca, nSottomarini, nBatiscafi)) : ( )

pre: -

post: Stampa a video le tuple dell'insieme c.

FineSpecificaAttivitàAtomica

Specifica attività I/O (3)

InizioSpecificaAttivitàAtomica StampaFine

StampaFine( ) : ( )

pre: -

post: Stampa a video il termine dell’esplorazione.

FineSpecificaAttivitàAtomica

Specifica attività atomiche (Task)

InizioSpecificaAttivitàAtomica AvviaEsplorazione

AvviaEsplorazione(e : Esplorazione) : ( )

pre: -

post:

invia segnali di InizioImmersione a tutte le Barche coinvolte in e.

FineSpecificaAttivitàAtomica

Specifica attività atomiche (Task) (2)

InizioSpecificaAttivitàAtomica InterrompiEsplorazione

InterrompiEsplorazione(e : Esplorazione) : ( )

pre: -

post:

invia segnali di Risalita ai Sottomarini contenuti nelle Barche coinvolte in e.

FineSpecificaAttivitàAtomica

Specifica attività atomiche (Task) (3)

InizioSpecificaAttivitàAtomica VerificaDisponibili

VerificaDisponibili(e : Esplorazione) :

(Insieme(Barca,nSottomarini,nBatiscafi)

pre:

post:

sia **b** una Barca coinvolta in e, **s** un sottomarino contenuto in **b** e **bat** un batiscafo incluso in **s**. La tupla <**b**, **ns**, **nb**> è composta dalla Barca **b**, e dagli interi ns e nb pari rispettivamente, al numero di Sottomarini **s** il cui stato sia *A_BORDO* e Batiscafi **bat** il cui stato sia *PRONTO.*

result è l'insieme di tutte le tuple <**b**, **ns**, **nv**>.

FineSpecificaAttivitàAtomica

Specifica attività complesse

InizioSpecificaAttività Esplora

Esplora(e : Esplorazione) : ( )

Variabili: -

Inizio Processo:

AvviaEsplorazione(e) : ( )

AttendiStop(e) : ( )

InterrompiEsplorazione(e) : ( )

Fine Processo

FineSpecificaAttività

Specifica attività complesse (2)

InizioSpecificaAttività Verifica

Verifica(e : Esplorazione) : ( )

Variabili:

conta : Insieme(Barca,nSottomarini,nBatiscafi

Inizio Processo:

VerificaDisponibili(e) : (conta)

StampaReport(conta) : ( )

Fine Processo

FineSpecificaAttività

Specifica attività complesse (3)

InizioSpecificaAttività AttivitaPrincipale

AttivitaPrincipale(e : Esplorazione) : ( )

Variabili: -

Inizio Processo:

fork:

t1 : {Esplora(e) : ( )}

t2 : {Verifica(e) : ( )}

join t1,t2.

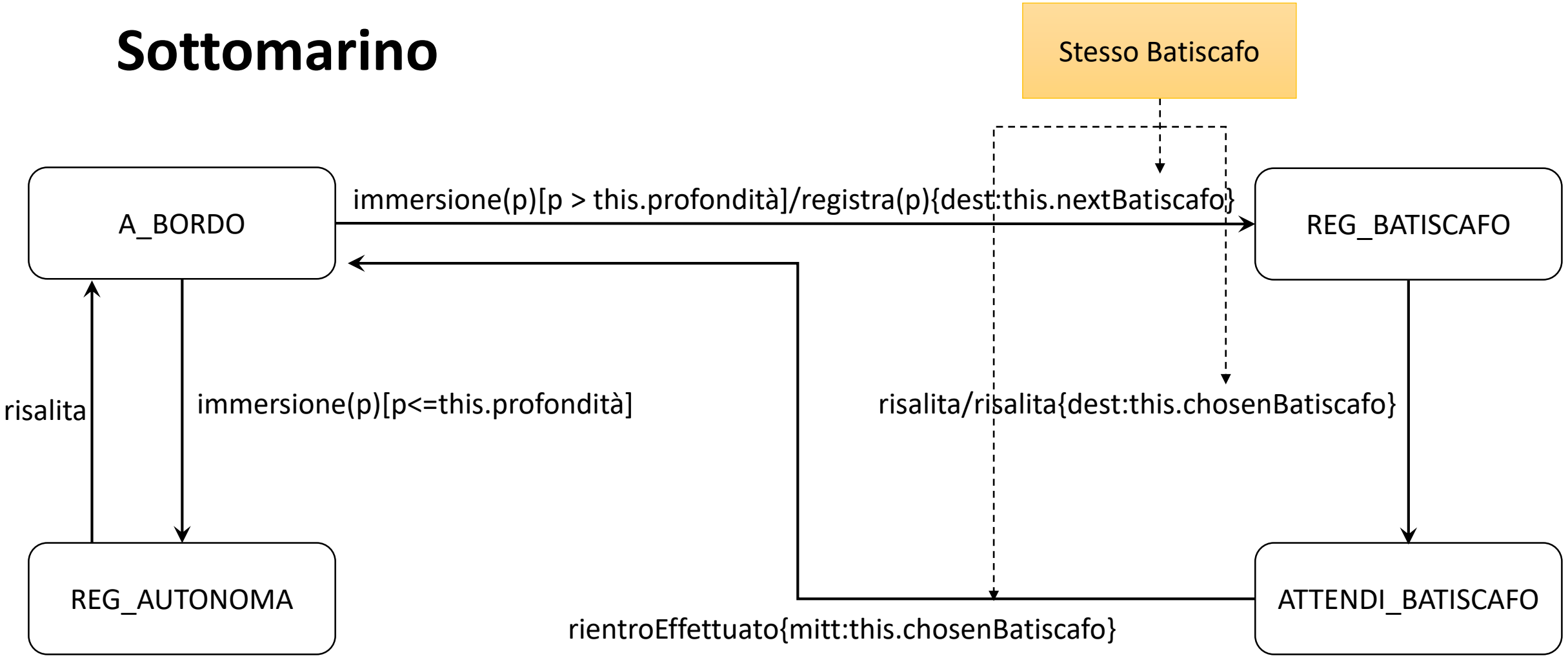
StampaFine( ) : ( )

Fine Processo.

FineSpecificaAttività

Fase di Analisi – Diagramma Stati e Transizioni

Sottomarino



Specifica Stati - Sottomarino

InizioSpecificaStatiClasse Sottomarino

Stato: {A_BORDO, REG_AUTONOMA, REG_BATISCAFO,
ATTENDI_BATISCAFO}

Variabili di stato ausiliarie:

profonditàAttuale : int

chosenBatiscafo : Batiscafo

Stato Iniziale:

statoCorrente = A_BORDO

profonditàAttuale = 0

chosenBatiscafo = -

Fine Specifica

Specifica Transizioni – Sottomarino

InizioSpecificaTransizioneClasse Sottomarino

Transizione: A_BORDO -> REG_AUTONOMA

Evento: immersione(p : intero)

Condizione: p <= this.profondità

Azione:

pre: evento.mitt == this.contiene

post: this.profonditàAttuale = p

FineSpecifica

Specifica Transizioni – Sottomarino (2)

InizioSpecificaTransizioneClasse Sottomarino

Transizione: REG_AUTONOMA -> A_BORDO

Evento: risalita

Condizione:

Azione:

pre: evento.mitt == this.contiene

post: this.profonditàAttuale = 0

FineSpecifica

Specifica Transizioni – Sottomarino (3)

InizioSpecificaTransizioneClasse Sottomarino

Transizione: A_BORDO -> REG_BATISCAFO

Evento: immersione(p : intero)

Condizione: p > this.profondità

Azione:

pre: evento.mitt == this.contiene

post: this.profonditàAttuale = this.profondità

this.chosenBatiscafo = this.nextBatiscafo( )

nuovoevento = registra(p){dest: this.chosenBatiscafo}

FineSpecifica

Specifica Transizioni – Sottomarino (4)

InizioSpecificaTransizioneClasse Sottomarino

Transizione: REG_BATISCAFO -> ATTENDI_BATISCAFO

Evento: risalita

Condizione:

Azione:

pre: evento.mitt == this.contiene

post: *nuovoevento* = risalita{dest: this.chosenBatiscafo}

FineSpecifica

Specifica Transizioni – Sottomarino (5)

InizioSpecificaTransizioneClasse Sottomarino

Transizione: ATTENDI_BATISCAFO -> A_BORDO

Evento: rientroEffettuato

Condizione:

Azione:

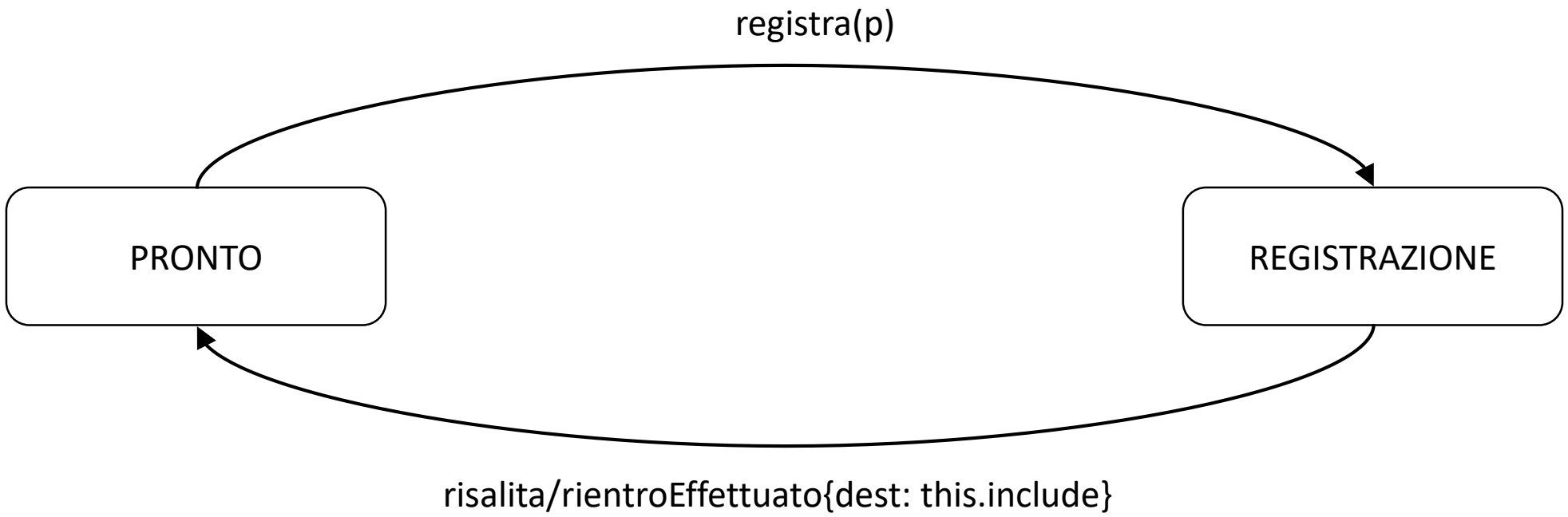
pre: evento.mitt == this.chosenBatiscafo

post: this.profondità = 0

FineSpecifica

Fase di Analisi – Diagramma Stati e Transizioni

Batiscafo



Specifica Stati - Batiscafo

InizioSpecificaStatiClasse Batiscafo

Stato: {PRONTO, REGISTRAZIONE}

Variabili di stato ausiliarie:

profonditàAttuale : int

Stato Iniziale:

statoCorrente = PRONTO

profonditàAttuale = 0

Fine Specifica

Specifica Transizioni – Batiscafo

InizioSpecificaTransizioneClasse Batiscafo

Transizione: PRONTO -> REGISTRAZIONE

Evento: registra(p : intero)

Condizione: -

Azione:

pre: evento.mitt == this.include

post: this.profonditàAttuale = p

FineSpecifica

Specifica Transizioni – Batiscafo (2)

InizioSpecificaTransizioneClasse Batiscafo

Transizione: REGISTRAZIONE -> PRONTO

Evento: risalita

Condizione: -

Azione:

pre: evento.mitt == this.include

post: profonditàAttuale = 0

nuovoevento = rientroEffettuato{dest: this.include}

FineSpecifica

Fase di Progetto – Responsabilità sulle Associazioni

(1) requisiti, (2) molteplicità, (3) operazioni.

Associazione	Classe	Responsabilità
associato	Dispositivo Esplorazione	NO SI (1,3)
riguarda	Esplorazione ZonaMare	SI (1, 2) NO
contiene	Barca Sottomarino	SI (2,3) SI (2)
include	Sottomarino Batiscafo	SI (2,3) SI (2)

Faso di Progetto – Strutture Dati

Abbiamo la necessità di rappresentare collezioni omogenee di oggetti, a causa:

- dei vincoli di molteplicità 1..* e 0..* delle associazioni,
- delle variabili necessarie per vari algoritmi

Per fare ciò, utilizzeremo le classi del collection framework di Java: Set, HashSet.

Fase di Progetto – Tabella corrispondenza tipi UML

Tipo UML	**Tipo JAVA**
intero	int
string	String
Insieme	Set
reale	double

Fase di Progetto – Tabelle di gestione delle proprietà

Classe UML	**Proprietà Immutabile**
Dispositivo	nome
Esplorazione	nome
ZonaMare	lat lon
Barca	-
Sottomarino	-
Batiscafo	-

Classe UML	**Proprietà**	
	Nota alla nascita	**Non nota alla nascita**
-	-	-

Fase di Progetto – Altre Considerazioni

Sequenza di nascita degli oggetti: Non dobbiamo assumere una particolare sequenza di nascita degli oggetti.

Valori default alla nascita: Non sembra ragionevole assumere che per qualche proprietà esistano valori di default validi per tutti gli oggetti.