

LISTA 15 – MODULARIZACAO – EXERCICIO RALLY

ALGORITMO ``RALLY``

FUNCAO leiaNumInteiroIntervalo(inf, sup: INTEIRO): INTEIRO

VAR

n: INTEIRO

INICIO

REPITA

 ESCREVA ("Digite valor no intervalo [``, inf, '','', sup ,'']: ``")

 LEIA n

 SE(n<inf) OU (n>sup) ENTAO

 ESCREVAL ("Fora do intervalo solicitado!! ")

 FIMSE

ATE (n>=inf) E (n<=sup)

RETORNE n

FIMFUNCAO

PROCEDIMENTO leiaTempos(VAR t1, t2, t3: INTEIRO)

INICIO

 t1 <- leiaNumInteiroIntervalo(1,180)

 t2 <- leiaNumInteiroIntervalo(1,180)

 t3 <- leiaNumInteiroIntervalo(1,180)

FIMPROCEDIMENTO

PROCEDIMENTO leiaEquipe(VAR inscr, a, b, c: INTEIRO)

INICIO

 ESCREVAL ("Inscricao")

 inscr <- leiaNumInteiroIntervalo(1,10)

 ESCREVAL ("Tempos equipe")

 leiaTempos(a, b, c)

FIMPROCEDIMENTO

FUNCAO calculaEtapa(tp, te: INTEIRO): REAL

VAR

 dif: INTEIRO

INICIO

 dif <- abs(tp-te)

 SE(dif<3)ENTAO

 RETORNE 100

 SENAO

 SE(dif<=5)ENTAO

 RETORNE 80

 SENAO

 RETORNE 80-(dif-5)/5.0

 FIMSE

 FIMSE

FIMFUNCAO

PROCEDIMENTO calculaPontos(tp1, tp2, tp3, tel, te2, te3: INTEIRO; VAR p1, p2, p3: REAL)

INICIO

 p1 <- calculaEtapa(tp1, tel);

 p2 <- calculaEtapa(tp2, te2);

 p3 <- calculaEtapa(tp3, te3);

FIMPROCEDIMENTO

PROCEDIMENTO escrevePontos(p1, p2, p3: REAL)

INICIO

 ESCREVAL("Pontuacoes: `` ,p1 , `` + `` ,p2,`` + `` ,p3)

 ESCREVAL("Total: `` , p1+p2+p3)

FIMPROCEDIMENTO

VAR

 tp1, tp2, tp3, tel, te2, te3, n, i, inscr, ganha:INTEIRO

 p1, p2, p3, tot, maior: REAL

INICIO

 ESCREVAL(``Tempos padrao")

 leiaTempos(tp1, tp2, tp3)

 ESCREVAL

```

ESCREVAL("Qtde de equipes ")
n <- leiaNumInteiroIntervalo(1,10)
PARA i DE 1 ATE n FACA
    ESCREVAL
    ESCREVAL(" Dados equipe:")
    leiaEquipe(inscr, te1, te2, te3)
    calculaPontos(tp1, tp2, tp3, te1, te2, te3, p1, p2, p3)
    escrevePontos(p1, p2, p3)
    tot <- p1+p2+p3
    SE( maior < tot) ENTAO
        maior <- tot
        ganha <- inscr
    FIMSE
FIMPARA
ESCREVAL
ESCREVAL( "Equipe vencedora: `` , ganha , `` - `` , maior , ``pontos")
FIMALGORITMO

```

LISTA 15 – MODULARIZACAO – EXERCICIO RALLY – EM C++

```
#include <iostream>
```

```

int leiaNumInteiroIntervalo(int inf, int sup) { // funcao
    int n;
    do{
        cout << "Digite valor no intervalo [`` << inf << ',' << sup << '']: ``";
        cin >> n;
        if(n<inf || n>sup)
            cout << "Fora do intervalo solicitado!! " << endl;
    } while (n<inf || n>sup);
    return n;
}

void leiaTempos(int& t1, int& t2, int& t3) { // procedimento c/param p/ref
    t1 = leiaNumInteiroIntervalo(1,180);
    t2 = leiaNumInteiroIntervalo(1,180);
    t3 = leiaNumInteiroIntervalo(1,180);
}

void leiaEquipe(int& inscr, int& a, int& b, int& c){ // proced c/param p/ref
    cout << "Inscricao";
    inscr = leiaNumInteiroIntervalo(1,10);
    cout << "Tempos equipe";
    leiaTempos(a, b, c);
}

float calculaEtapa(int tp, int te){ // funcao
    int dif = abs(tp-te);
    if(dif<3) return 100;
    if(dif<=5) return 80;
    return 80-(dif-5)/5.0;
}

// procedimento c/param p/valor e p/ref
void calculaPontos(int tp1, int tp2, int tp3, int te1, int te2, int te3, float&
p1, float& p2, float& p3){
    p1 = calculaEtapa(tp1, te1);
    p2 = calculaEtapa(tp2, te2);
    p3 = calculaEtapa(tp3, te3);
}

// proced classico param p/valor
void escrevePontos(float p1, float p2, float p3){
    cout << "Pontuacoes: `` <<p1 <<`` + ``<<p2<<`` + `` <<p3 << endl;
    cout << "Total: ``<< p1+p2+p3 << endl;
}

int main(){
    int tp1, tp2, tp3, te1, te2, te3, n, i, inscr, ganha;

```

```

float p1, p2, p3, tot, maior=0;

cout<<"Tempos padrao"<<endl;
leiaTempos(tp1, tp2, tp3);
cout<<"\nQtde de equipes "<<endl;
n = leiaNumInteiroIntervalo(1,10);
for(i=1; i<=n; i++){
    cout<<"\nDados equipe:"<<endl;
    leiaEquipe(inscr, te1, te2, te3);
    calculaPontos(tp1, tp2, tp3, te1, te2, te3, p1, p2, p3);
    escrevePontos(p1, p2, p3);
    tot = p1+p2+p3;
    if( maior < tot) {
        maior = tot;
        ganha = inscr;
    }
}
cout << "\n\nEquipe vencedora: ``<< ganha << `` - `` << maior << ``pontos";
return 1;
}

```