9.

A afirmação é verdadeira

O polimorfismo ocorre quando se declara numa classe base um membro-funcão como virtual.

Isto faz com que, ao chamar a funcão, se nas classes derivadas existir um membro-funcao com igual definição, será este membro-funcao

a ser invocado, não o da classe base.

Por exemplo:

class Pessoa{

/\*...\*/

public:

virtual void imprimeInformacao(); //imprime para a consola

};

class Estudante: public Pessoa{

/\*...\*/

public:

void imprimeInformacao();

};

---------------------------------------

int main(){

vector<Pessoa \*> v

preencheVectorComPessoas(v); //funcao que preence um vector do tipo Pessoa \* com apontadores para pessoas aleatorias

Estudante\* es;

es->preencheCampos(); //preenche a pessoa localizada em es com dados aleatorios

v.push\_back(es);

v.at(v.size()-1)->imprimeInformacao(); //imprime a informação do estudando usando o membro-funcao da classe Estudante

}

--------------------------------------------

Assim, com o polimorfismo, consegue-se a adaptar a invocacao de um membro-funcao de acordo com o contexto pretendido.

10.

Numa pesquisa binária, é necessario garantir que o vector esteja ordenado antes de se começar a pesquisa

Esta ordenação prévia do vector pode, dependendo do seu tamanho e do algoritmo utilizado, anular a rapidez

caracteristica da pesquisa binária, podendo até torna-lo mais lento que o de pesquisa sequencial

Porém, só é necessário ordenar o vector uma unica vez. Sendo assim, se for preciso fazer várias pesquisas, a desvantagem de gastar recursos

para ordenar o vector começa a desvanecer, ficando mais eficaz, a longo prazo, utilizar a pesquisa binaria.

Resumindo, a pesquisa binária, apesar de ser eficaz ( O( log(N) ) ) tem a desvantagem de se ter de ordenar o vector, enquanto a

pesquisa sequencial, sendo menos rapida (O(N)), não acarreta essa necessidade.