

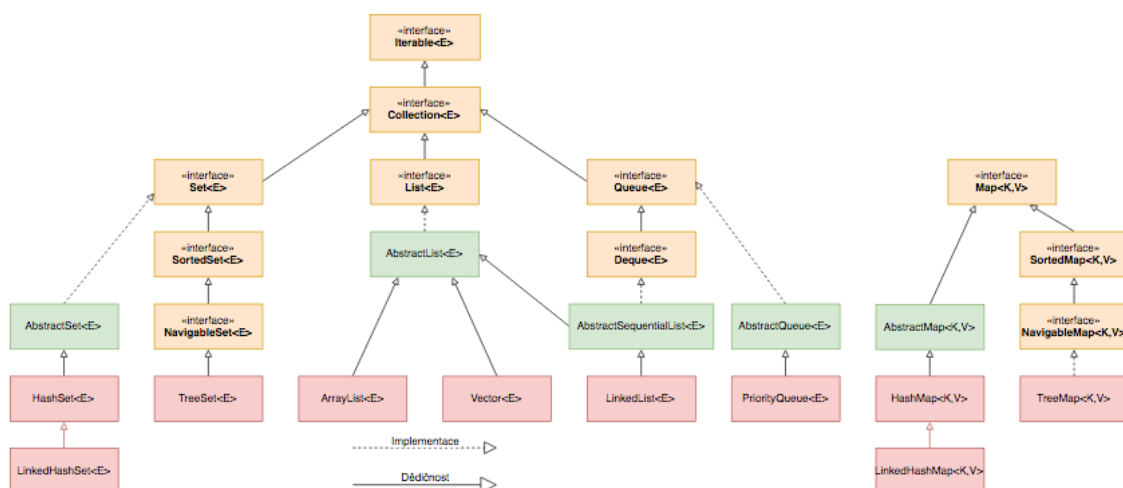
TRABALHO FINAL

INFORMAÇÕES GERAIS:

- Esta atividade versa sobre a *Java Collections API* e compreende 8 (oito) questões.
- Algumas questões devem ser respondidas através de um texto argumentativo com a justificativa da resposta.
- Nas questões que exigem implementação, descrever os algoritmos utilizados e explicar como executar a aplicação.
- Escrever um relatório técnico no formato pdf (máximo 5 páginas) com a solução das questões. O trabalho deve ser apresentado oralmente pelo grupo.
- Alternativamente, o grupo pode explorar as *Collections* da linguagem Python: *Container datatypes*, desde que seja acordado com o professor. Python possui *containers* embutidos: *dict*, *list*, *set* e *tuple*. Python 3 implementou *containers* especializados: *namedtuple*, *deque*, *ChainMap*, *Counter*, *OrderedDict*, *defaultDict*, *UserDict*, *UserList*, *UserString*.

CONSIDERAÇÕES:

Collections em Java são usadas para armazenar grupos de objetos. A *Collections API* fornece um número de interfaces (incluindo *Collection*, *List*, *Map* e *Set*) para definir uma maneira padrão de usar uma faixa concreta de estrutura de dados. As interfaces e classes da *Collections API* pertencem ao pacote *java.util* e são mostradas na figura abaixo:



QUESTÕES:

1. Descrever as principais características das classes *Vector*, *LinkedList* e *ArrayList*, incluindo vantagens e desvantagens. Fazer uma aplicação Java para criar uma tabela de símbolos usando cada uma dessas classes para armazenar as palavras do arquivo “*leipzig100k.txt*” que possui cerca de 140.000 palavras. As palavras normalmente são separadas com o caractere *<space>*, *<tab>* e *<newline>*, através do método *split()*. No entanto, podem ser utilizados outros delimitadores como: dígitos e sinais de pontuação. Qual o tempo total de execução de cada uma das classes implementadas?
2. Descrever as principais características da classe *HashSet*, *LinkedHashSet* e *TreeSet*, incluindo vantagens e desvantagens. Fazer uma aplicação Java para criar uma tabela de símbolos usando cada uma dessas classes para armazenar as palavras do arquivo “*leipzig100k.txt*” que possui cerca de 140.000 palavras. As palavras normalmente são separadas com o caractere *<space>*, *<tab>* e *<newline>*, através do método *split()*. No entanto, podem ser utilizados outros delimitadores como: dígitos e sinais de pontuação. Qual o tempo total de execução de cada uma das classes implementadas?
3. Descrever as principais características da classe *HashMap*, *LinkedHashMap* e *TreeMap*, incluindo vantagens e desvantagens. Fazer uma aplicação Java para criar uma tabela de símbolos usando cada uma dessas classes para armazenar as palavras do arquivo “*leipzig100k.txt*” que possui cerca de 140.000 palavras. As palavras normalmente são separadas com o caractere *<space>*, *<tab>* e *<newline>*, através do método *split()*. No entanto, podem ser utilizados outros delimitadores como: dígitos e sinais de pontuação. Qual o tempo total de execução de cada uma das classes implementadas? Os valores associados com as chaves do *Map* podem ser desconsiderados (i.e., valor *zero* ou *null*).
4. Fazer um gráfico com o tempo de execução das aplicações criadas nas questões anteriores, para avaliar a operação de inclusão nas diversas estruturas.
5. Repetir as questões de 1 a 3 e, após criar a tabela de símbolos, implementar um trecho de código para consultar 10 palavras, a saber: Lisbon, NASA, Kyunghee, Konkuk, Sogang, momentarily, rubella, vaccinations, government, Authorities.
6. Fazer um gráfico com o tempo de execução das aplicações criadas na questão anterior, para avaliar a operação de busca de dados nas diversas estruturas.
7. Repetir as questões de 1 a 3 e, após criar a tabela de símbolos, implementar um trecho de código para excluir 10 palavras, a saber: Lisbon, NASA, Kyunghee, Konkuk, Sogang, momentarily, rubella, vaccinations, government, Authorities.
8. Fazer um gráfico com o tempo de execução das aplicações criadas na questão anterior, para avaliar a operação de remoção de dados nas diversas estruturas.

OBS: Nas avaliações temporais, tentar ser justo e usar sempre o mesmo computador, além de considerar o tempo médio de pelo menos cinco execuções.