Übung 9: Hashing

Abgabetermin: 05.06.2018

Name: **Matrikelnummer:**

Gruppe: ☐ **G1** Di 10:15-11:00 ~~☐ **G2** Di 11:00-11:45~~ ☐ **G3** Di 10:15-11:00

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	8	☐	Java-Programm	Projekt Archiv	☐	
Aufgabe 2	8	☐	Java-Programm	Projekt Archiv	☐	
Aufgabe 3	8	☐	Java-Programm	Projekt Archiv	☐	

Aufgabe 1: Hashtabelle mit linearem Probieren (8 Punkte)

Implementieren Sie eine Hashtabelle mit der Kollisionsstrategie “linear probing” in der Klasse *HashMapLinearProbing*. Die Schnittstelle ist durch die abstrakte Klasse Map gegeben.

Abzugeben ist: Projekt Archiv

Aufgabe 2: Hashtabelle mit quadratischem Probieren (8 Punkte)

Implementieren Sie eine Hashtabelle mit der Kollisionsstrategie “quadratic probing” in der Klasse *HashMapQuadraticProbing.* Berücksichtigen Sie den in der Vorlesung beschriebenen Trick, um bei der Suche alle Tabellenplätze zu besuchen. Die Schnittstelle ist durch die abstrakte Klasse Map gegeben.

Abzugeben ist: Projekt Archiv

Aufgabe 3: Hashtabelle mit “Separate Chaining” (8 Punkte)

Implementieren Sie eine Hashtabelle zur Assoziation von Object-Schlüsseln mit Object-Werten mit der Kollisionsstrategie “separate chaining” in der Klasse *HashMapSeparateChaining*. Verwenden Sie die Methode *hashCode* um den Hash für die Schlüssel zu berechnen. Die Schnittstelle ist durch die abstrakte Klasse *Map* gegeben.

Abzugeben ist: Projekt Archiv

Implementierungshinweise:

- Verwenden Sie das Vorgabeprojekt **PI2_UE09.zip**.
- Fügen Sie Ihre Implementierung in den mit **TODO** markierten Teilen den Klassen *HashMapSeparateChaining, HashMapLinearProbing* und *HashMapQuadraticProbing* ein.
- Verwenden Sie *PrimeNumbers.calculateTabSize(int)* im Paket *at.jku.ssw.util* um eine geeignete Tabellengröße für *HashMapQuadraticProbing* zu errechnen.
- Benutzen Sie die Methode *HashMapUtil.mod(a, b)* um sicher zu stellen, dass der Rest positiv ist. (In Java ist das Ergebnis von `a % b` negativ, wenn `a` negativ ist.)
- Markieren Sie gelöschte Einträge in *HashMapLinearProbing* und *HashMapQuadraticProbing* mit *Pair.DELETED_MARK* um die Suchkette aufrecht zu erhalten.
- Beachten Sie, dass die Map voll werden kann. Werfen Sie in diesem Fall in den *put* Methoden eine IllegalStateException.
- Ändern sie **keine public Interfaces** vorgegebener Skeleton Klassen.
- Halten Sie sich an die **Codierungsrichtlinien** auf der Kurs Website.