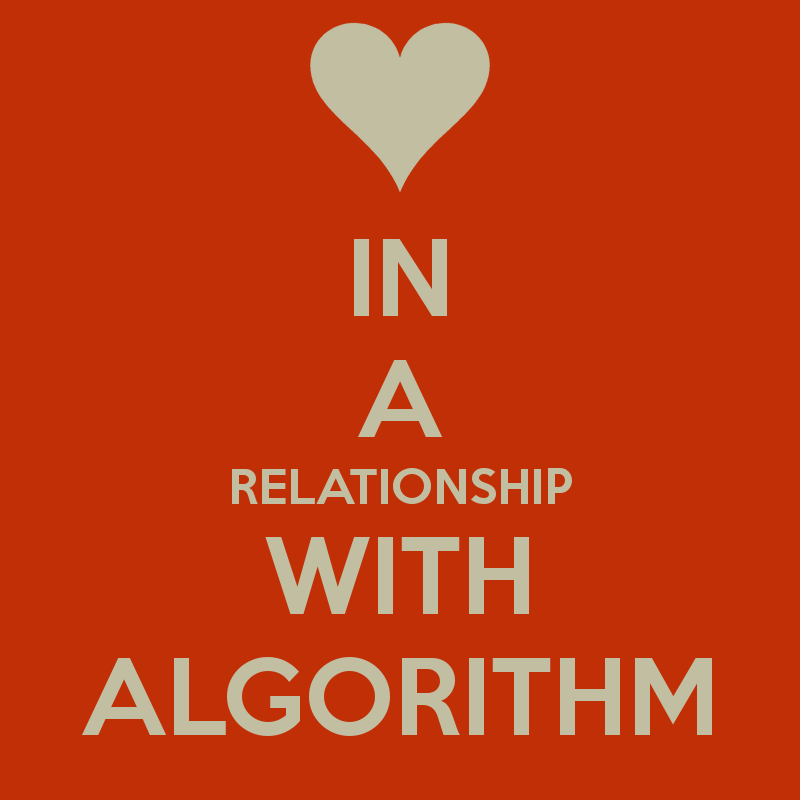
**Datastructure**

**Practicum2B**



Klas: IS201

Naam: Jeffrey, Alwin

Inhoudsopgave

**Inleiding2**

**Class Diagram3**

**Insertion Sort**4

**Bucket Sort**5

**Test**7

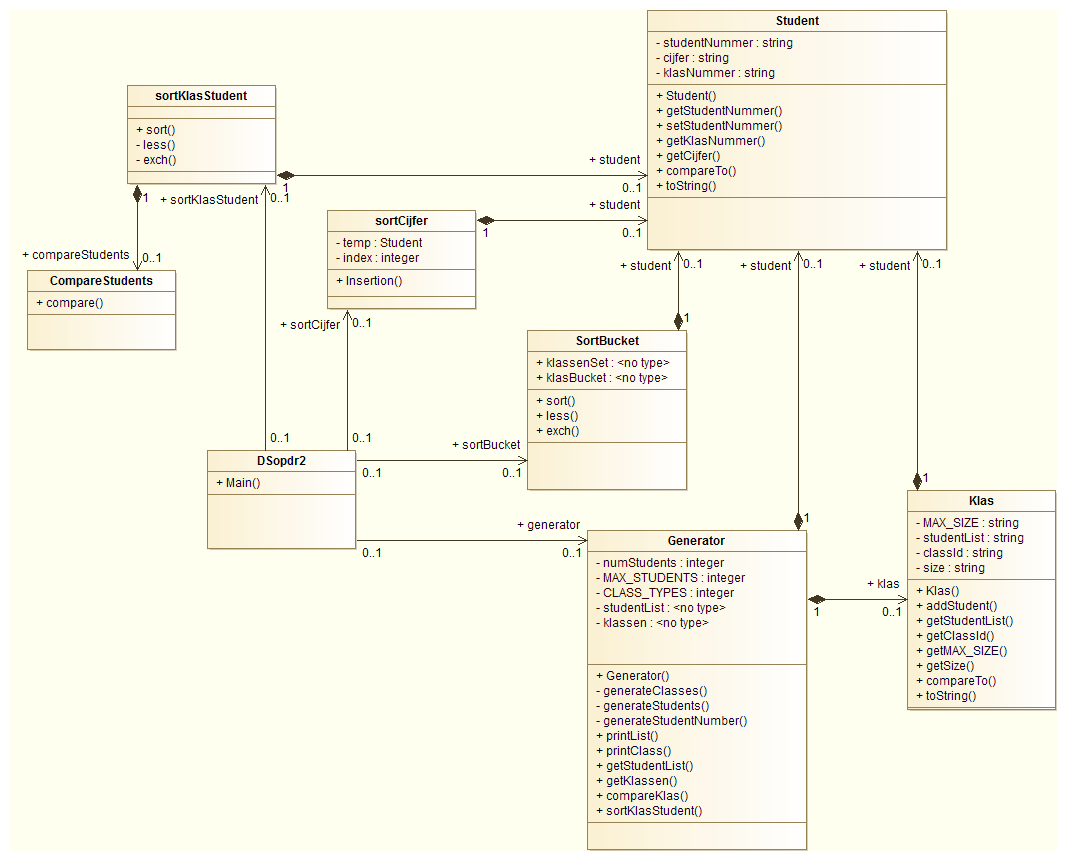
**Conclusie**8

**Inleiding**

We hebben een applicatie geschreven die een lijst met een aantal van 200, 400, 800, 1600 en 3200 studenten moet kunnen generen. Deze leerlingen krijgen allemaal een unieke studentennummer, met een cijfer van 1 tot 10 en een klas naam waarin de student uiteindelijk wordt ingedeeld. Er zijn 4 klassen richtingen IG, IS, IN of IT. Het applicatie moet een leerlingen lijst kunnen generen met behulp van een insertion sort gesorteerd op cijfers van 10 naar 1 van hoog naar laag. En het applicatie moet een lijst studenten genereren met een bucket sort gesorteerd op klassen in alfabetische volgorde en vervolgens sorteren op studentennummer per klas.

In het verslag kom 10 test resultaten van de insertion en bucket sort van toepassing, met daarbij de uiteindelijke conclusie.

**Class Diagram**



**Insertion Code:**

temp(student) vergeleken met de vorige student in compareTo method van de Class Student.

Als temp hoger is dan verplaats het vorig student een index hoger, 'index' wordt steeds lager en de getallen worden steeds hoger totdat temp een getal tegen komt die hoger is dan hem zelf.

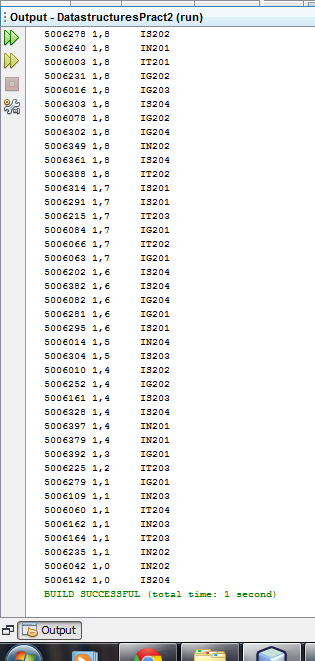
|  |
| --- |
| public class sortCijfer {    private static Student temp = null;  private static int index = 0;    public static void Insertion(ArrayList<Student> sort){    for (int i = 0; i < sort.size() ; i++) {  temp = sort.get(i);  index = i;    while (index > 0 && temp.compareTo(sort.get(index - 1)) > 0)  {  sort.set(index, sort.get(index - 1));  index --;  }  sort.set(index, temp);  }  }  } |

compareTo vergelijkt 2 cijfers van 2 verschillende studenten met elkaar.

Als het een 10 heeft of tegen komt wordt het tijdelijk vervangen met een final String om verwarring tussen 10 en 1 te voorkomen.

|  |
| --- |
| @Override  public int compareTo(Student s) {  final String temp = "99";    if(s.getCijfer().equals("10"))  return cijfer.compareTo(temp);  if(this.cijfer.equals("10"))  return temp.compareTo(s.getCijfer());    return cijfer.compareTo(s.getCijfer());  } |

Insertion sort Resultaat:



## Test Efficiency

Bucket sort resultaat:

