Termin: Montag, 12. Mai 2014

Abschlussprüfung Sommer 2014

Aufgabenbogen

Mathematisch-technischer Softwareentwickler
Mathematisch-technische Softwareentwicklerin
6511

4 Entwicklung eines Softwaresystems

3 Phasen, davon

- 7 Stunden schriftliche Aufgabe
- 4 Tage Realisierung des Konzepts
- 30 Minuten Fachgespräch

100 Punkte

Parkettroboter

Die Firma Plan & Eben entwickelt einen Roboter, der automatisch eine Parkettfläche versiegeln soll. Die zu bearbeitende Fläche soll rechteckig sein und aus Parzellen der Größe 40 cm x 40 cm bestehen. Zur Einschränkung der Komplexität soll die Fläche auf maximal 10 Parzellen in jeder Richtung beschränkt sein. Sie kann Hindernisse aufweisen, die vom Roboter umfahren werden müssen.

Prinzip der Verarbeitung:

- Im Rahmen der Vorbereitung soll von einem Programm beginnend von einem gegebenen Startpunkt ein Weg über die Parkettfläche ermittelt werden, auf dem die Versiegelung der Fläche möglich ist. Dieser Weg liegt dann in Form einer Liste von anzufahrenden Parzellen (dem Routenplan) vor.
- Erst nach Bestimmung und Laden des Routenplans wird der Roboter auf die vorher ausgewählte Start-Parzelle (erster Punkt in der Liste) gestellt und eingeschaltet. Anschließend fährt das Gerät – ausgehend von der Start-Parzelle – der Reihe nach die Positionen im geladenen Routenplan an. Alle auf dem so definierten Weg überfahrenen Parzellen werden dabei ohne Unterbrechung versiegelt.
- Die Verarbeitung endet, wenn die letzte Parzelle aus dem Routenplan erreicht (und damit versiegelt) ist. Das Gerät bleibt dort stehen und schaltet sich automatisch aus.

Der Roboter:

- Der Roboter hat die vier Bewegungsrichtungen „nach oben", „nach rechts", „nach unten" und „nach links". Diese Bewegungsrichtungen gelten absolut – unabhängig von der aktuellen Ausrichtung des Gerätes.
- Der Roboter fährt der Reihe nach die im Routenplan aufgeführten Parzellen an und versiegelt sie dabei.
- Über der zuletzt bearbeiteten Parzelle bleibt das Gerät stehen und schaltet sich nach der Versiegelung aus.

Randbedingungen für Ermittlung des Routenplans:

- Der Roboter darf sich ausgehend von einer Parzelle, die gerade verarbeitet wurde, in jede der verfügbaren Bewegungsrichtungen auf eine andere, noch nicht verarbeitete Parzelle bewegen.
- Dabei dürfen allerdings bereits versiegelte Parzellen nicht erneut überfahren werden; weil die Versiegelung erst belastbar aushärten muss.
- Ebenso dürfen natürlich die Parzellen mit den Hindernissen nicht angefahren werden.
- Am Ende der Verarbeitung darf er über der letzten versiegelten Parzelle stehen bleiben.

Beispiel 1:

Die Parkettfläche umfasst 5 mal 5 Parzellen. Die Start-Parzelle ist an der Position (1,1). Die Parkettfläche enthält keine Hindernisse.

Die Eingabe für das Programm (Kommentarzeilen beginnen mit einem Semikolon):

```
; Abmessungen der Flaeche
5 5
; Startparzelle
1 1
; Eckparzellen der Hindernisse, ein Hindernis pro Zeile
; (in diesem Fall ohne Hindernisse)
```

Die Eingabe spezifiziert zunächst die Größe der Fläche (hier 5 Reihen hoch und 5 Spalten breit) und die Position der Start-Parzelle (1,1). Es folgen hier keine Spezifikationen für Hindernisse. Das Dateiende schließt die Eingabe ab.

Die Ausgabe für das Programm:

```
Startposition (1,1)
   1 2 3 4 5
 1 S
 2
 3
 4
 5

Uhrzeiger-Strategie
   1 2 3 4 5
 1 > > > > V
 2 V < V < V
 3 V ^ V ^ V
 4 V ^ V ^ V
 5 Z ^ < ^ <
```

```
Routenplan:
1,1 / 1,5 / 5,5 / 5,4 / 2,4 / 2,3 / 5,3 / 5,2 / 2,2 / 2,1 / 5,1

zu versiegelnde Parzellen:   25
Hindernisparzellen:          0
versiegelte Parzellen:       25
nicht versiegelte Parzellen: 0
```

Die Ausgabe soll sich an diesem Beispiel orientieren und somit in folgender Reihenfolge geschehen:

- Ausgabe der Startposition
- Ausgabe des Routenplans für jede Strategie zur Routensuche (vgl. Aufgabenstellung)
  - als rechteckiges Muster in ASCII-Form und
  - als Liste
- Ausgabe einer Kurzzusammenfassung der Routensuche

Dabei symbolisieren die Zeichen „^", „>", „v", bzw. „<" die Fahrtrichtungen „nach oben", „nach rechts", „nach unten" bzw. „nach links". Damit lässt sich der spätere Weg des Roboters über die Fläche verfolgen. Auf der letzten Parzelle (5,1) symbolisiert das „Z" die Endposition.

Hindernisse:

Die Parkettfläche kann zusätzlich ein oder mehrere rechteckige Hindernisse enthalten, die auch nebeneinander liegen dürfen. Diese werden bei der Verarbeitung ausgespart, d. h. deren Grundfläche ist nicht zu bearbeiten. Außerdem müssen sie natürlich bei den späteren Fahrbewegungen des Roboters vermieden werden. Die Grundfläche der Hindernisse passt genau in das angesprochene Raster der Parkettfläche. Sie kann also aus einer oder mehreren Parzellen der Größe 40 cm x 40 cm bestehen.

Bei der Planung der Fahrbewegungen des Roboters müssen also nicht nur die frisch versiegelten Parzellen, sondern auch diese Hindernisse berücksichtigt werden.

Die Hindernisse werden in der Eingabedatei zeilenweise durch Angabe diagonal gegenüberliegender Eckpunkte spezifiziert und folgen auf die Zeilen mit der Angabe der Größe der Fläche und des Startfeldes.

In der Ausgabe werden Parzellen mit Hindernissen mit ‚H' gekennzeichnet.

Zielvorstellung:

Die Firma Plan & Eben benötigt einem Algorithmus (und die entsprechende Implementierung in einer objektorientierten Programmiersprache), der bei vorgegebener Topologie einer Parkettfläche (Außenmaße, Start-Parzelle und Hindernisse) einen Routenplan für den Roboter derart ermittelt, dass möglichst die gesamte Fläche verarbeitet werden kann.

Es soll also ein möglicher unterbrechungsfreier Verarbeitungsweg ermittelt oder aber festgestellt werden, dass die Verarbeitung nicht unterbrechungsfrei zu erledigen ist. Ein solcher Weg – wenn er denn existiert – soll ermittelt und in Form eines Routenplans ausgegeben werden.

Vorschläge zur Ermittlung des Routenplans:

Ausgehend von einer Position des Roboters, gibt es theoretisch vier Möglichkeiten, den Roboter zu bewegen:

a) im Plan nach oben

b) im Plan nach rechts

c) im Plan nach unten

d) im Plan nach links

Bei der Entscheidung für eine dieser Richtungen müssen natürlich die Hindernisse und die bereits versiegelten Parzellen berücksichtigt werden. Deshalb fällt in den allermeisten Fällen mindestens eine dieser Möglichkeiten aus (nämlich mindestens die Richtung, aus der die aktuelle Position angesteuert wurde).

Für die Strategie, nach der die nächste anzufahrende Parzelle bestimmt wird, gibt es sicherlich viele Varianten. Hier wird zunächst eine mögliche Variante, die Uhrzeiger-Strategie, vorgestellt.

Uhrzeiger-Strategie:

1. Eine mögliche **Strategie** bei der konkreten Entscheidung, auf welches Feld als nächstes gefahren werden soll, ist z. B. die folgende:
   - Auf jeder Parzelle wird die Entscheidung der nächsten zu wählenden Parzelle im Uhrzeigersinn beginnend mit „im Plan nach oben" getroffen. Sollte keine dieser vier Richtungen sinnvoll sein, so endet der Weg auf dieser Parzelle.
   - Eine Bewegung in die getestete Richtung ist dann sinnvoll, wenn die dortige Parzelle sich noch auf der zu versiegelnden Parkettfläche befindet, kein Hindernis enthält und auf dem bisherigen Weg noch nicht versiegelt wurde.
2. Nach Erreichen eines Endpunktes (d. h. wenn es keine direkt erreichbare Parzelle mehr gibt) wird geprüft, ob auf dem aktuellen Weg alle Parzellen angefahren und damit bearbeitet wurden.

   Ist dies der Fall, stoppt der Algorithmus, denn es soll lediglich ein zulässiger Verarbeitungsweg gesucht werden.

   Werden auf dem aktuellen Weg nicht alle Felder angefahren, muss geprüft werden, ob durch eine andere (frühere) Richtungs-Entscheidung im Verlauf des Weges ggf. ein zulässiger und vollständiger Weg ermittelt werden kann.

Eine weitere Strategie nach Wahl:

Neben der Uhrzeiger-Strategie ist eine weitere Strategie zur Suche des Verarbeitungsweges zu entwickeln und zu implementieren.

Es folgt ein weiteres Beispiel mit einem Hindernis.

Beispiel 2:
Die Eingabe für das Programm:

```
Eingabe:
; Abmessungen der Flaeche
5  6
; Startparzelle
5  1
; Eckparzellen der Hindernisse, ein Hindernis pro Zeile
2 4 5 4
```

Die Ausgabe für das Programm:

```
Startposition (5,1)
   1 2 3 4 5 6
 1
 2       H
 3       H
 4       H
 5 S     H

Uhrzeiger-Strategie:
   1 2 3 4 5 6
 1 > V > > > V
 2 ^ V ^ H Z V
 3 ^ V ^ H ^ V
 4 ^ V ^ H ^ V
 5 ^ > ^ H ^ <

Routenplan:
5,1 / 1,1 / 1,2 / 5,2 / 5,3 / 1,3 / 1,6 / 5,6 / 5,5 / 2,5

< an dieser Stelle soll der Routenplan in ASCII- und Listenform für eine zweite Strategie ausgegeben werden, vgl. Aufgabenstellung >

zu versiegelnde Parzellen:   26
Hindernisparzellen:          4
versiegelte Parzellen:       26
nicht versiegelte Parzellen: 0
```

Beispiel 3:

In diesem Beispiel ist es nicht möglich, alle Parzellen zu versiegeln. In diesem Fall soll – vgl. Aufgabenstellung – ein Weg bestimmt werden, auf dem möglichst viele Parzellen versiegelt werden.

Die Eingabe für das Programm:

```
;  Abmessungen der Fläche
  3 5
;  Startparzelle
  2 1
;  Eckparzellen der Hindernisse, ein Hindernis pro Zeile
  2 3   2 5
```

Die Ausgabe für das Programm:

```
Startposition (2,1)
   1 2 3 4 5
 1
 2 S   H H H
 3

Uhrzeiger-Strategie:
   1 2 3 4 5
 1 > V
 2 ^ V H H H
 3   > > > Z

Routenplan:
2,1 / 1,1 / 1,2 / 3,2 / 3,5

< an dieser Stelle soll der Routenplan in ASCII- und Listenform für eine zweite Strategie ausge-
geben werden, vgl. Aufgabenstellung >

zu versiegelnde Parzellen:   12
Hindernisparzellen:          3
versiegelte Parzellen:       8
nicht versiegelte Parzellen: 4
```

bitte wenden!

Aufgabenstellung:

Schreiben Sie ein Programm, das zu einer gegebenen Parkettfläche und zu einer gegebenen Startposition einen Routenplan für den Parkettroboter ermittelt und ausgibt, der zu einer vollständigen Versiegelung der zu versiegelnden Fläche führt. Falls kein solcher vollständiger Routenplan existiert, soll einer der unvollständigen Routenpläne ausgegeben werden, der die meistmöglichen Parzellen versiegelt.

Implementieren Sie zwei deterministische Strategien zur Parzellenauswahl in Ihrem Programm:

1.) Die Uhrzeiger-Strategie.

2.) Eine andere, sich von der Uhrzeiger-Strategie wesentlich unterscheidende Strategie zur Routensuche.

Testen Sie Ihre Testbeispiele mit beiden Strategien, geben Sie für beide Strategien den Endzustand und den Routenplan aus und diskutieren Sie die Ergebnisse.

Im Rahmen der schriftlichen Aufgabe sind am ersten Tag abzugeben:

- Aufgabenanalyse und der Entwurf des Algorithmus zur Suche eines Routenplans gemäß beider Strategien
- Programmkonzeption unter Berücksichtigung der funktionalen Trennung
  - o Klassen, Methoden und Datenstrukturen in Form von UML-Diagrammen
  - o UML-Sequenzdiagramme für die wesentlichen Abläufe
  - o Detaillierte Beschreibung der wesentlichen Methoden in Form von Nassi-Shneiderman-Diagrammen

Im Rahmen des Prüfprodukts sind in gedruckter und elektronischer Form abzugeben:

- Verbale Beschreibung der realisierten Verfahren
- Programmsystem (bestehend aus Klassen, Schnittstellen, Methoden) als Quellcode
- Entwicklerdokumentation
- Benutzeranleitung
- ausführliche Beschreibung, Begründung und Diskussion
  - o der angegebenen Beispiele und
  - o einer ausreichenden Zahl von Beispielen mit Ein- und Ausgabe, die sowohl die Normalfälle als auch auftretende Spezial- und Fehlerfälle abdecken
- Zusammenfassung und Ausblick (z. B. Erweiterungsmöglichkeiten)

Im Rahmen des Prüfprodukts sind in elektronischer Form abzugeben:

- Programmsystem in ausführbarer Form
- Ein- und Ausgabedateien Ihrer Beispiele
- ein Skript zur automatischen Ausführung aller von Ihnen angegebenen Beispiele

Dem Prüfprodukt ist eine eigenhändig unterschriebene Eigenhändigkeitserklärung (Juristische Erklärung) folgenden Inhalts beizufügen:

„Ich erkläre verbindlich, dass das vorliegende Prüfprodukt von mir selbstständig erstellt wurde. Die als Arbeitshilfe genutzten Unterlagen sind in der Arbeit vollständig aufgeführt. Ich versichere, dass der vorgelegte Ausdruck mit dem Inhalt des von mir erstellten Datenträgers identisch ist. Weder ganz noch in Teilen wurde die Arbeit bereits als Prüfungsleistung vorgelegt. Mir ist bewusst, dass jedes Zuwiderhandeln als Täuschungsversuch zu gelten hat, der die Anerkennung des Prüfprodukts als Prüfungsleistung ausschließt."

Im Rahmen des auftragsbezogenen Fachgesprächs sind die Aufgabenanalyse und der Lösungsentwurf zu begründen und das Prüfungsprodukt zu erläutern.