

Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel

Tim Schulte, Dr. Christian Becker-Asano, Dr. Stefan Wölfl

Wintersemester 2014/2015

Universität Freiburg

Institut für Informatik

Übungsblatt 9

Abgabe: Freitag, 9. Januar 2015, 18:00 Uhr

WICHTIGE HINWEISE: Zur Bearbeitung der Übungsaufgaben legen Sie bitte ein neues Unterverzeichnis `sheet09` im Wurzelverzeichnis Ihrer Arbeitskopie des SVN-Repositories an. Ihre Lösungen werden dann entsprechend dem ersten Übungsblatt in Dateien in diesem Unterverzeichnis erwartet. Beachten Sie bitte bei allen Aufgaben die *Hinweise zur Bearbeitung der Übungsaufgaben* unter der folgenden URL:

http://gki.informatik.uni-freiburg.de/teaching/ws1415/info1/hinweise_uebungen.txt

Insbesondere müssen alle Python-Dateien und die darin definierten Funktionen entsprechend diesen Hinweisen dokumentiert und dabei PEP8 beachtet werden.

Unterbinden Sie alle nicht erforderlichen `print`-Anweisungen in Ihren Python-Dateien. Falls Sie zum Debuggen `print`-Anweisungen verwenden, benutzen Sie eine globale Variable oder Konstante, mit der Sie Ausgaben auf die Konsole ausschalten können.

Überprüfen Sie, dass Sie alle Lösungen ins Repository hochgeladen haben. Bewertet wird bei allen Aufgaben die letzte Version, die zur Deadline des Übungsblattes auf dem SVN-Server eingereicht ist.

Aufgabe 9.1 (Ethische Reflektion, Punkte: 4, Datei: `ethics.txt`)

Wie beurteilen Sie die folgende Situation unter ethischen Gesichtspunkten? Ist Tims Entscheidung in der geschilderten Situation ethisch vertretbar (oder gar ethisch geboten)? Wenn nicht, was sollten die Personen stattdessen tun?

Wichtig ist, dass Sie Ihre Beurteilung ausreichend begründen. Ziehen Sie dazu die Leitlinien des VDI und der GI sowie den *Software Engineering Code of Ethics and Professional Practice* (<http://www.acm.org/about/se-code>) der ACM zu Rate.

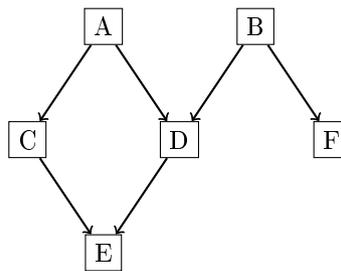
Tim arbeitet bei einer kleineren Firma, die Software-Module für sicherheitskritische Komponenten in der Automobil-Industrie entwickelt. Die Spezifikationen dieser Software-Module werden vom jeweiligen Auftraggeber vorgegeben. Tim weiß, dass die Auftraggeber die gelieferte Software üblicherweise nochmals umfangreichen eigenen Tests unterziehen.

Kurz vor dem nächsten Software-Release zeigt das Modul, an dem Tim mit anderen Kollegen arbeitet, in bestimmten Testkonfigurationen Laufzeitfehler. Welche Auswirkungen dies im Verbund anderer Software-Komponenten hat, kann Tim nicht abschätzen. Auch die Spezifikation des Auftraggebers ist für diese Konfigurationen nicht eindeutig. Weil Tim den Fehler nicht lokalisieren kann, wendet er sich an seinen Chef und teilt ihm das Problem mit. Dieser weist ihn an, die Laufzeitfehler auf den Testkonfigurationen so abzufangen, dass ein einwandfreies Verhalten zumindest

auf den Test-Instanzen zugesichert ist. Eine tiefer gehende Fehleranalyse und damit eine Verschiebung des Release-Termins lehnt er kategorisch ab: bereits beim letzten Release sei der Zeitrahmen überschritten worden und der Auftraggeber habe bei weiteren Verspätungen Vertragsstrafen angekündigt, die die kleine Firma empfindlich treffen würden.

Tim entscheidet sich dazu, der Anweisung seines Vorgesetzten Folge zu leisten.

Aufgabe 9.2 (MRO, 4 Punkte, Datei: `mro.txt`)



Wenden Sie auf die obige Klassenhierarchie die in der 18. Vorlesung eingeführte `merge`-Funktion schrittweise an, um die Method-Resolution-Order für die Unterklasse E zu bestimmen.

Aufgabe 9.3 (RoboRally, Punkte 2+4+4, Datei: `roborally.py`)

In dieser Aufgabe beschäftigen wir uns mit der Implementierung weiterer `FactoryElements` für das in der Vorlesung vorgestellte Brettspiel *RoboRally*. Sie erweitern dazu die Datei `roborally.py`, nachdem Sie diese von der Vorlesungswebseite heruntergeladen haben:

<http://www.informatik.uni-freiburg.de/~ki/teaching/ws1415/info1/python/roborally-aufgabe.py>

In dieser Datei sind die Klassen und die zu bestehenden Doctests bereits angegeben. Sie sollten den bestehenden Quellcode nur dann ändern, wenn Sie diese Änderungen gut begründen können und sie Ihnen zwingend notwendig erscheinen. Ihre konkrete Aufgabe ist es, die beiden Fabrikelemente „Schrottpresse“ und „Einweg-Portal“ als neue Klassen zu implementieren:

Schrottpresse (Crusher): Falls die Presse *aktiv* ist (während der angegebenen Registerphasen), wird der Roboter, der auf diesem Feld steht, zerstört. Dies geschieht im 5. Arbeitsschritt (`elem_move`).

Einweg-Portal (OneWayPortal): Ein Einweg-Portal schickt einen Roboter von seiner momentanen Position zu einer anderen, bei der Initialisierung des Portals angegebenen, Position. Dies geschieht im 6. Arbeitsschritt. Bei Kollisionen wird ähnlich wie bei Förderbändern vorgegangen: Alle Roboter werden parallel bewegt. Existiert am Ende eine Kollision, werden die Bewegungen der kollidierten Roboter zurück abgewickelt. Beachten Sie mögliche unerwünschte Interaktionen mit Wänden!

Die Aufgaben:

1. Überlegen und begründen Sie, wie die zwei neuen Fabrikelemente in die bestehende Klassenhierarchie (siehe auch Folie 28 der Vorlesung `infoI18.pdf`) einzuordnen sind. Wovon sollten `Crusher` und `OneWayPortal` jeweils sinnvoller Weise abgeleitet werden und warum?
2. Implementieren Sie eine neue Klasse `Crusher`, die die oben beschriebene „Schrottpresse“ realisiert. Beachten Sie, dass Schrottpressen nur in bestimmten Registerphasen aktiv sind.
3. Implementieren Sie eine neue Klasse `OneWayPortal`, die das oben beschriebene „Einweg-Portal“ realisiert.

Hinweis: Unter Umständen müssen sie auch eine weitere Methode für `MoveableThings` implementieren. Beachten Sie mögliche unerwünschte Interaktionen mit Wänden! Diese lassen sich durch eine Anpassung der `Wall`-Klasse beseitigen.

Aufgabe 9.4 (Erfahrungen; 2 Punkte)

Legen Sie im Unterverzeichnis `sheet09` eine Textdatei `erfahrung.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, benötigter Zeitaufwand nach Teilaufgabe, Bezug zur Vorlesung, Interessantes, etc.).

*Wir wünschen Ihnen ein frohes Weihnachtsfest
und ein gutes Neues Jahr 2015.*