

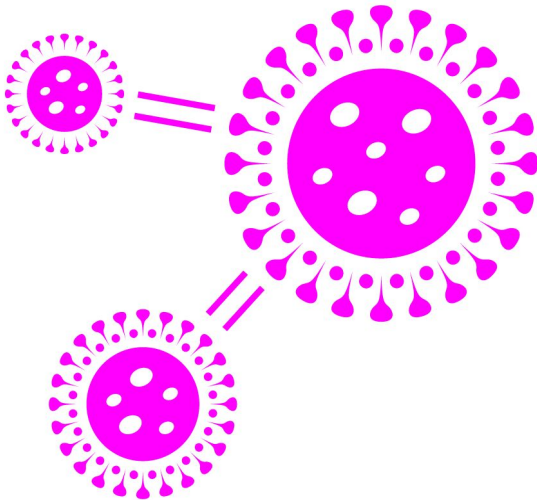
## Pandemie!

Keime sind Mikroorganismen oder subzelluläre Erreger, die in anderen Organismen pathogene (d.h. gesundheitsschädigende) Abläufe verursachen. Solche Krankheitserreger können zum Beispiel Bakterien oder Viren sein. Die Ansteckung mit einem Krankheitserreger nennt man Infektion. Voraussetzung für eine Infektion ist die Ausbreitung der Keime, so dass diese mit einem Organismus in Kontakt kommen. Im Organismus erfolgt dann eine Vermehrung als Grundlage für eine weitere Ausbreitung.

Für den Schutz gegen Keime sowie für deren Bekämpfung existieren eine Reihe von möglichen Maßnahmen. Bei einer Quarantäne werden infizierte Organismen isoliert, um die weitere Ausbreitung von Keimen zu verhindern. Impfungen aktivieren das Immunsystem gegen spezifische Keime und beugen so Infektionen vor. Arzneimittel wie zum Beispiel Virostatika oder Antibiotika hemmen die Vermehrung von Keimen oder töten diese sogar ab.

Viele gefährliche Krankheiten wie Cholera, Ebola oder Masern erfordern die effiziente Auswahl und effektive Anwendung möglicher Gegenmaßnahmen. Dieser anspruchsvollen Aufgabe widmet sich der diesjährige InformatiCup.

## Aufgabe



Implementieren Sie eine Software, die in einem rundenbasierten Spiel die Menschheit vor der Auslöschung durch eine Pandemie<sup>1</sup> rettet.

Gegeben ist eine Menge von Städten. Eigenschaften der Städte mit unveränderlichen Werten sind Name, Koordinaten und Flugverbindungen zu anderen Städten. Eigenschaften mit veränderlichen Werten sind Einwohnerzahl, Stärke der Wirtschaft, Stabilität der Regierung, Hygienestandards und Achtsamkeit der Einwohner.

---

<sup>1</sup> <https://de.wikipedia.org/wiki/Pandemie>

Zu Beginn des Spiels brechen in zufällig gewählten Städten zufällig gewählte Keime aus. Keime haben folgende Eigenschaften, deren Werte alle unveränderlich sind.

- **Name:** Fiktive Bezeichnung des Keims
- **Infektiosität:** Wie hoch ist die Wahrscheinlichkeit, dass ein Infizierter andere Einwohner seiner Stadt oder Einwohner verbundener Städte infiziert?
- **Mobilität:** Wie wahrscheinlich ist der Übergang des Keims auf nicht verbundene Städte?
- **Dauer:** Wie viele Runden dauert die Infektion? Ein Infizierter ist während der gesamten Dauer ansteckend.
- **Letalität:** Wie hoch ist die Wahrscheinlichkeit, dass ein Infizierter stirbt? Infizierte sterben entweder oder werden immun und können nicht mehr mit dem Keim infiziert werden.

Die Menge der Keime ist endlich und Ihnen zunächst nicht bekannt. Nach hinreichend vielen Spielen werden Sie alle Keime gesehen haben. Eigenschaften der Städte und Keime wirken sich auf die Ausbreitung der Keime innerhalb der Städte und zwischen den Städten sowie auf die Eintrittswahrscheinlichkeiten von Ereignissen aus.



Ausbreitung und Auswirkungen der Keime basieren auf einem fiktiven, stark vereinfachtem, stochastischen Modell. Bitte verwenden Sie kein Biologie-Lehrbuch für die Lösung dieser Aufgabe sondern ausschließlich die Ergebnisse Ihrer Beobachtungen innerhalb der Spiele.

In jeder Runde können Sie beliebig viele Aktionen durchführen (siehe Abschnitt *Ausgabe*). Jede Aktion kostet Punkte. Sie beginnen ein Spiel mit 40 Punkten Startguthaben und erhalten pro Runde 20 weitere Punkte. Das Spiel endet, wenn alle Keime eliminiert wurden oder die Menschheit ausgelöscht wurde. Die Menschheit gilt als ausgelöscht, wenn am Ende einer beliebigen Runde die Summe der Einwohner aller Städte gegenüber der ersten Runde (mehr als) halbiert wurde.

## Eingabe

Ihre Software muss einen Webservice bereitstellen, über den die Eingabe der Spielzustände per HTTP-POST-Requests erfolgt. Im Body der Requests sind Objekte in JSON<sup>2</sup> (UTF-8) wie in dem folgenden Beispiel enthalten.

Die numerischen Eigenschaftswerte der Städte und Keime werden auf die Angaben "--" (sehr niedrig), "-", "o", "+" und "++" (sehr hoch) abgebildet.

---

<sup>2</sup> [https://de.wikipedia.org/wiki/JavaScript\\_Object\\_Notation](https://de.wikipedia.org/wiki/JavaScript_Object_Notation)

```

{
  "round": 4, // Aktuelle Runde, zu Beginn des Spiels 1
  "outcome": "pending", // Ausgang des Spiels, "win" | "loss" | "pending"
  "points": 20, // Verfügbare Punkte, natürliche ganze Zahl
  "cities": { // Mindestens 2 Einträge
    "Hamburg": { // Name der Stadt, eindeutig über alle Städte
      "latitude": 53.548450, // Längengrad
      "longitude": 9.978514, // Breitengrad
      "population": 1822, // 10^3 Einwohner, natürliche ganze Zahl
      "connections": ["Berlin", "Köln", "München", ...], // Flugverbindungen
      "economy": "++", // Stärke der Wirtschaft
      "government": "+", // Stabilität der Regierung
      "hygiene": "++", // Hygienestandards
      "awareness": "+", // Achtsamkeit der Einwohner
      "events": [ // Ereignisse (siehe den Hinweis zu Ereignistypen)
        {
          "type": "outbreak", // Ausbruch eines Keims
          "sinceRound": "4", // Runde, in der das Ereignis eingetreten ist
          "pathogen": {
            "name": "Coccus innocuus", // Name des Keims
            "infectivity": "+", // Infektiosität
            "mobility": "-", // Mobilität
            "duration": "-", // Dauer
            "lethality": "--", // Letalität
          },
          "prevalence": 0.34, // Anteil der infizierten Einwohner der Stadt
        },
        {
          "type": "uprising", // Aufstand!
          "sinceRound": "4", // Runde, in der das Ereignis eingetreten ist
          "participants": 36, // 10^3 Teilnehmer, natürliche ganze Zahl
        }
      ]
    },
    "Berlin": { ... }, ...
  ],
  // Siehe Felder "events" der Städte. Diese Ereignisse betreffen das gesamte
  // Spiel und nicht nur einzelne Städte.
  "events": [ ... ],
  // Feld "error" ist nur enthalten, wenn eine Aktion nicht akzeptiert wurde.
  "error": "..." // Natürlichsprachliche Beschreibung
}

```

Spielzustände können zu unterschiedlichen Spielen gehören. Da sie nicht eindeutig einem Spiel zugeordnet werden können, muss Ihre Software in der Lage sein, Entscheidungen ohne Rückgriff auf vorige Spielzustände zu treffen.

Die Menge der Ereignistypen ist (analog zur Menge der Keime) endlich und Ihnen zunächst unbekannt. Dies umfasst die unterschiedlichen Eigenschaften der Ereignistypen und deren Auswirkungen auf das Spielgeschehen. Manche Ereignisse lassen Rückschlüsse auf vorige Spielzustände zu, beispielsweise seit welcher Runde ein Impfstoff gegen einen Keim entwickelt wird.

Berücksichtigen Sie die Umsetzung der Produktivversion Ihrer Software als zustandslosen Webservice von Anfang an in Ihrem Softwareentwurf.

## Ausgabe

Auf eine Eingabe des Spielstandes muss Ihr Webservice mit einem Aktions-Objekt in JSON (UTF-8) wie in dem folgenden Beispiel antworten.

```
{
  "type": "closeAirport", // Aktionstyp laut Tabelle
  "city": "Berlin", // Sonstige Felder laut Tabelle
  "rounds": 4
}
```

Im Folgenden finden Sie eine vollständige Aufstellung der Ihnen zur Verfügung stehenden Aktionen.

Name	Format	Kosten (Punkte)
Runde beenden	<code>{"type": "endRound"}</code>	0
Die aktuelle Runde wird beendet. Es können in dieser Runde keine weiteren Aktionen ausgeführt werden. Der nächste Spielzustand wird eingegeben.		
Quarantäne anordnen	<code>{"type": "putUnderQuarantine", "city": "&lt;Name einer Stadt: S&gt;", "rounds": "&lt;Anzahl Runden: R, natürliche Zahl &gt; 0&gt;"}</code>	$10 \times \text{Anzahl Runden} + 20$
Keime breiten sich in den nächsten $R$ Runden nicht von $S$ in andere Städte aus.		
Flughafen schließen	<code>{"type": "closeAirport", "city": "&lt;Name einer Stadt: S&gt;", "rounds": "&lt;Anzahl Runden: R, natürliche Zahl &gt; 0&gt;"}</code>	$5 \times \text{Anzahl Runden} + 15$
Keime breiten sich in den nächsten $R$ Runden nicht über Flugverbindungen von und nach $S$ aus.		
Flugverbindung sperren	<code>{"type": "closeConnection", "fromCity": "&lt;Name einer Stadt: S1&gt;", "toCity": "&lt;Name einer Stadt: S2&gt;", "rounds": "&lt;Anzahl Runden: R, natürliche Zahl &gt; 0&gt;"}</code>	$3 \times \text{Anzahl Runden} + 3$
Keime breiten sich in den nächsten $R$ Runden nicht über die Flugverbindung von $S1$ nach $S2$ aus.		
Impfstoff entwickeln	<code>{"type": "developVaccine", "pathogen": "&lt;Name eines Keims: K&gt;"}</code>	40
In der 6. Runde nach Ausführung dieser Aktion steht ein Impfstoff gegen den Keim $K$ zur Verfügung. Voraussetzung: $K$ muss im Laufe des Spiels bereits ausgebrochen sein.		
Impfstoff verteilen	<code>{"type": "deployVaccine", "pathogen": "&lt;Name eines Keims: K&gt;", "city": "&lt;Name einer Stadt: S&gt;"}</code>	5
Alle nicht infizierten Einwohner der Stadt $S$ werden sofort immun gegen den Keim $K$ . Voraussetzung: Ein Impfstoff gegen den Keim $K$ steht zur Verfügung.		

Medikament entwickeln	<code>{"type": "developMedication", "pathogen": "&lt;Name eines Keims: K&gt;"}</code>	20
In der 3. Runde nach Ausführung dieser Aktion steht ein Medikament gegen den Keim <i>K</i> zur Verfügung. Voraussetzung: Der Keim <i>K</i> muss im Laufe des Spiels ausgebrochen sein.		
Medikament verteilen	<code>{"type": "deployMedication", "pathogen": "&lt;Name eines Keims: K&gt;", "city": "&lt;Name einer Stadt: S&gt;"}</code>	10
30% bis 50% der mit dem Keim <i>K</i> infizierten Einwohner der Stadt <i>S</i> werden geheilt und sind in der Folge immun gegen <i>K</i> . Voraussetzung: Ein Medikament gegen den Keim <i>K</i> steht zur Verfügung.		
Politischen Einfluss geltend machen	<code>{"type": "exertInfluence", "city": "&lt;Name einer Stadt: S&gt;"}</code>	3
Der Eigenschaftswert <i>Stärke der Wirtschaft</i> der Stadt <i>S</i> wird zufällig neu gesetzt.		
Neuwahlen ausrufen	<code>{"type": "callElections", "city": "&lt;Name einer Stadt: S&gt;"}</code>	3
Der Eigenschaftswert <i>Stabilität der Regierung</i> der Stadt <i>S</i> wird zufällig neu gesetzt.		
Hygienemaßnahmen durchführen	<code>{"type": "applyHygienicMeasures", "city": "&lt;Name einer Stadt: S&gt;"}</code>	3
Eigenschaftswert <i>Hygienestandards</i> der Stadt <i>S</i> wird zufällig erhöht.		
Informationskampagne starten	<code>{"type": "launchCampaign", "city": "&lt;Name einer Stadt: S&gt;"}</code>	3
Der Eigenschaftswert <i>Achtsamkeit der Einwohner</i> der Stadt <i>S</i> wird zufällig erhöht.		

## Testen

Im GitHub-Repository des aktuellen Wettbewerbs<sup>3</sup> finden Sie ein Kommandozeilenwerkzeug<sup>4</sup> als ausführbare Datei für Windows, macOS und Linux (x86-64). Mit Hilfe dieses Werkzeugs können Sie Ihre Implementierung testen. Alle Funktionen des Kommandozeilenwerkzeugs können Sie mit dem Parameter `--help` erfragen. Im GitHub-Repository finden Sie auch eine Beispiellösung, die Ihnen als Orientierungshilfe dienen kann. Die Beispiellösung stellt einen Webservice bereit, der zu jeder Eingabe des Spielstandes die Aktion *Runde beenden* ausgibt.

<sup>3</sup> <https://github.com/InformatiCup/InformatiCup2020>

<sup>4</sup> <https://github.com/InformatiCup/InformatiCup2020/releases/latest>

## Deployment

Die Juroren werden Ihre Software als Webservice verwenden. Dazu müssen Sie Ihre Software bitte entsprechend wie folgt unmittelbar bau- oder ausführbar bereitstellen.

1. Sie stellen ein baubares Dockerfile zur Verfügung oder (XOR)
2. Sie stellen einen ausführbaren Webservice auf einer Plattform Ihrer Wahl zur Verfügung.



Eine Beschreibung wie Sie mit AWS Lambda<sup>5</sup> deployen können finden Sie hier:

<https://docs.aws.amazon.com/lambda/latest/dg/deploying-lambda-apps.html>



Das Deployment Ihrer Software ist in diesem Jahr ein Teil der Aufgabenstellung. Die entsprechende Verfügbarkeit Ihrer Lösung ist Voraussetzung für deren Bewertung.

Die Juroren werden Ihre Software mit dem unter *Testen* beschriebenen Kommandozeilenwerkzeug testen.

## Bewertung

Ihre Software sollte innerhalb möglichst weniger Runden alle Keime eliminieren. Lösung **A** wird höher bewertet als Lösung **B**, wenn...

- **A** alle Keime eliminiert und bei **B** die Menschheit ausgelöscht wird.
- **A** und **B** beide alle Keime eliminieren und **A** weniger Runden benötigt.
- Bei **A** und **B** die Menschheit ausgelöscht wird und bei **A** dafür mehr Runden benötigt werden.

Neben der Software erstellen Sie bitte eine Ausarbeitung, die die Installation und Bedienung Ihrer Software sowie Ihren theoretischen Lösungsansatz beschreibt. Ihre Einreichung wird ganzheitlich bewertet. Neben der Güte der Lösung werden der theoretische Lösungsansatz, die Form der Ausarbeitung, die Softwarearchitektur und -qualität, mögliche Erweiterungen und, wenn Sie Ihre Lösung im Finale vorstellen dürfen, die Qualität der Präsentation bewertet.



Im Anhang finden Sie eine Checkliste der Bewertungskriterien. Nutzen Sie diese Liste, um die Vollständigkeit Ihrer Lösung zu überprüfen.

## Außerdem

Die FAQs zu dieser Aufgabenstellung finden Sie in Kürze online in dem GitHub-Repository <https://github.com/InformatiCup/InformatiCup2020>.

---

<sup>5</sup> [https://en.wikipedia.org/wiki/AWS\\_Lambda](https://en.wikipedia.org/wiki/AWS_Lambda)

# Checkliste Bewertungskriterien

## Theoretischer Ansatz

Der theoretische Ansatz muss in einer Ausarbeitung, die zusammen mit der Implementierung eingereicht wird, dargestellt werden. Bewertet werden sowohl der Inhalt als auch die Form.

## Theoretische Ausarbeitung

- ☐ **Hintergrund:** Welche theoretischen Ansätze wurden verwendet? Warum wurden diese Ansätze verwendet?
- ☐ **Auswertung:** Wie hoch ist die Güte der Lösung?
- ☐ **Diskussion:** Wie lässt sich die Güte der Lösung erklären und erhöhen?
- ☐ **Quellen:** Wurden wissenschaftliche Quellen richtig und angemessen verwendet?

## Formalien

Eine gute Form ist entscheidend für die Lesbarkeit einer Ausarbeitung. Beachten Sie deshalb neben dem reinen Inhalt Ihrer Ausarbeitung bitte auch einige Formalien.

- ☐ **Rechtschreibung:** Rechtschreibung und Grammatik sind korrekt.
- ☐ **Struktur:** Eine klar erkennbare Struktur wird konsequent verfolgt.
- ☐ **Layout:** Das Dokument hat ein einheitliches Layout. Dieses kann frei gewählt werden, darf aber nicht den Lesefluss stören.
- ☐ **Zitate:** Es wird richtig und einheitlich zitiert.
- ☐ **Quellenangaben:** Quellen sind richtig und einheitlich angegeben.

## Softwarearchitektur und -qualität

Da eine etablierte Softwarearchitektur nur mit hohem Aufwand zu ändern ist, sollte sie besonders gründlich durchdacht und ausführlich begründet werden. Ausgewählte Aspekte der Softwarequalität sind für die Bewertung von besonderer Bedeutung. Gerne dürfen auch hier nicht genannte Aspekte aus den sehr weiten Feldern Softwarearchitektur und -qualität beleuchtet werden.

- ☐ **Architektur:** Beschreibung der Komponenten und deren Beziehungen
- ☐ **Software testing**<sup>6</sup>: Begründetes Konzept, Umsetzung
- ☐ **Coding conventions**<sup>7</sup>: Begründetes Konzept, Umsetzung
- ☐ **Wartbarkeit:** Mit welchem Aufwand kann das System angepasst werden?

## Handbuch

Das Handbuch beschreibt den Installationsprozess der Lösung als Webservice, insbesondere die Abhängigkeiten. Zudem wird die Benutzung erläutert. Empfohlen wird die genaue Angabe der erforderlichen Befehle oder die Bereitstellung eines Installations-Skripts.

## Erweiterungen

Erweitern Sie Ihre Lösung über die Anforderungen der Aufgabenstellung hinaus. Seien Sie kreativ!

## Präsentation

Im InformatiCup-Finale werden die besten Lösungen vor einer Fachjury präsentiert.

- ☐ **Foliendesign:** Sind die Folien ansprechend? Lenken sie nicht vom Inhalt der Präsentation ab?
- ☐ **Vortragsstil:** Weckt der Vortragsstil Interesse an der Präsentation?
- ☐ **Verständlichkeit:** Ist der Vortrag verständlich? Wird der Hintergrund der Lösung in einem angemessenen Tempo erklärt? Wird nötiges Vorwissen geschaffen?
- ☐ **Reaktion auf Nachfragen:** Können Nachfragen beantwortet werden?

---

<sup>6</sup> [https://en.wikipedia.org/wiki/Software\\_testing](https://en.wikipedia.org/wiki/Software_testing)

<sup>7</sup> [https://en.wikipedia.org/wiki/Coding\\_conventions](https://en.wikipedia.org/wiki/Coding_conventions)