**Part 2: Draw the memory map when the program runs. [3 points]**

Explain step by step what happened when the program runs and answer some questions.

* What is stored in the static heap, stack, dynamic heap?
* Why the Organization class is abstract?
* Why must the Colony/University class implement the communicateBytool() method?
* You explain the polymorphism feature in your code.
* Describe the difference between an interface and an abstract class.

**Explanation:**

The program ran smoothly until it reached the DrinkFood df line. However, since there was no DrinkFood class defined, the program crashed.

**Question:**

* **What is stored in the static heap, stack, dynamic heap?**

**Answer:**

**Stack:** hold data for running method.

* Stack frame for running method.
* Reference vari ables (e.g. obj1, obj2, df, …).
* Primitive variables in method.
* Running method (e.g. main(), …).

**Static heap**: where class definitions, static variables, and method bytecode are stored, shared across all instances of a class.

* Class definitions (e.g. Organization, Colony, BeeColony, University, FPTUniversity, …).
* Interface (e.g. Role, …).
* Static variables, constants, global primitive variables.
* Method definition (e.g. main(), toString(), …).

**Dynamic heap:** where objects (instances of classes) are created and stored at runtime, allowing for dynamic memory allocation.

* Objects (e.g. new BeeColony(2000, "honey", "land"), …) after new method
* Dynamic memory.
* **Why the Organization class is abstract?**

**Answer:**

* Because it contains abstract method communicateByTool()
* **Why must the Colony/University class implement the communicateBytool() method?**

**Answer:**

* Since they are sub-class of the Organization class and Organization class is an abstract class, they have to override and implement their parent abstract method.
* **You explain the polymorphism feature in your code.**

**Answer:**

* **The same method call behaves differently depending on the object** that is calling it.
* E.g. in Colony, the method communicationByTool() print out "the colony communicate by sound". But in University, the same method print out "in the university, people communicate by voice"

🡺same method behaves differently in diffent classes when implementing.

* **Describe the difference between an interface and an abstract class.**

**Answer:**

* Both **abstract classes** and **interfaces** contain abstract methods for subclasses to implement. The key differences are:
* **State Management**: Abstract classes can have **fields** and **behaviors** (methods) that hold values and share functionality with subclasses, similar to regular classes.
* **Method Implementations**: Abstract classes can include both **abstract** and **concrete methods**, while interfaces primarily define a **contract** for methods, with optional **default methods** for implementations since Java 8.