Exercise Session Informatik III

10. Chess Exercise

What the Hell is Chess?

- A **Board** with 8x8 black-and-white **Squares**
- Two **Players**
- Two differently colored sets of **Figures** each containing: 8 **Pawns**, 2 **Rooks**, 2 **Knights**, 2 **Bishops**, 1 **King** and 1 **Queen**.
- Rules describing legal **Moves**
- A ticking **Clock** to make the players nervous

globis

Exercise Tasks

- **Class Identification**
  Identify the main classes of the system and produce a class diagram.
- **Class Design**
  For each of the classes of your design, declare Eiffel classes in terms of features and assertions. Remember also to include comments!
- **Taking it Further**
  Assume that players want to be able to train with the system by viewing replays of games and by being able to undo and redo moves. How would you extend your design to allow for this?

globis

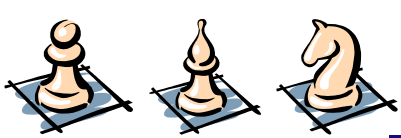
Focus of the Exercise

- Learn how to identify the **necessary classes** of an object-oriented system
- Learn how to design a **sensible, usable and stable interface**
- Learn how to design a system to be **extensible and flexible**
- The exercise is **not about implementing** code in Eiffel

globis

Class Identification

- What **elements** are essential to chess?
- Have some of these elements **common properties**?
- Are there different **variations** of one element?



globis

Class Design

- What **features** does the class need?
- Can **all desired funtionality** be achieved by the offered routines
- Are there redundant routines or are there **multiple ways** of getting one result?
- Who should be able to **access** each of the features?
- What **invariants** must hold?
- Are there enough **commentaries** for someone else to understand the class?

globis

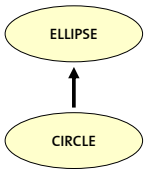
Taking it Further

- What concepts have to be **intoduced** to allow replays of games?
- Do these concepts in any way **affect the current design** of the application?
- Redo and undo has been covered in the lecture. What concept could be used in the chess environment to **mirror** the class COMMAND presented by Bertrand Meyer?

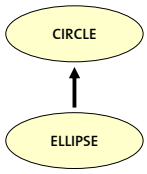
globis

Pitfalls of Class Design

- Is a **circle** an **ellipse** or is an **ellipse** a **circle**?



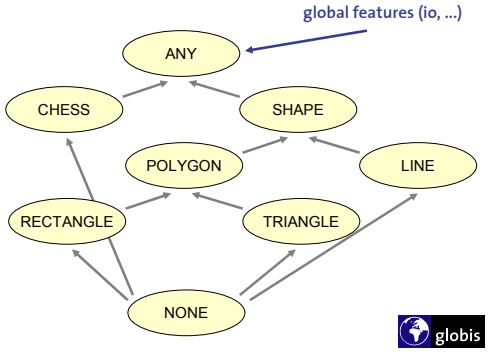
An instance of class **CIRCLE** is a certain kind of **ELLIPSE** where both axes have the same length, so **CIRCLE** is a specialisation of **ELLIPSE**.



An instance of class **ELLIPSE** has more features than a **CIRCLE**, so it is an extension of a **CIRCLE.**

globis

Eiffel Inheritance Graph



globis

That's all folks!

Bis zum nächsten Mal:
- Tasks 1 und 2 bearbeiten
- Diskussion
- Heute abend FIGUGEGL!

globis