Some system is characterized by a finite set of discrete states, which can be represented by integers in the range from -2^31 to 2^31. The states in the system can be characterized by: one integer, consecutive integers or scattered randomly. To model these three types of states, three classes have been created: “DiscreteState” - a single state, “SegmentState” - consecutive states, and “setState” - arbitrarily arranged states. The external events occurring can be described as random numbers in the range from -2^31 to 2^31. If a random number coincides with one of the states of the system, then the system enters an excited state.

To determine the probability of a system transition to an excited state, a class was constructed containing three overloaded methods: one for a parameter of the type “DiscreteState”, the second for “SegmentState”, the third for “setState". This led to code duplication. Later, it became necessary to add new entities to the model: a set of continuous states with omissions, a set of continuous states with additions, a set of continuous states with omissions and additions, a state as a union of two other states and a state as an intersection of two other states. Further code support based on overloaded methods was deemed impractical.

Suggest a solution based on inheritance, extension, and redefinition that will reduce the amount of code. Create classes for all the state variants described above in the model. Based on the received code, answer the following questions:

1. How does the probability of a random number hitting a set of points depend on the number of tests? (it is required to present dependency graphs for consecutive numbers and for arbitrarily arranged numbers)
2. How do the probability of a number falling into the set calculated by tests and the number of numbers in the set relate to the number of numbers in the test interval? (present sets of outcomes)
3. Does the probability calculated from the tests depend on the type of the set of states? (to present an analysis of points 1,2 for different types of sets of states)

The codes of the above classes can be found below: